

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»
Факультет инфокоммуникационных технологий

**ОТЧЕТ
О ЛАБОРАТОРНОЙ РАБОТЕ № 2**

по теме:

*«Запросы на выборку и модификацию данных, представления и
индексы в PostgreSQL»*

по дисциплине: Проектирование и реализация баз данных

Специальность:

45.03.04 Интеллектуальные системы в гуманитарной сфере

Проверила:

Говорова М.М.

Дата: «..» ... 2023 г.

Оценка _____

Выполнил:

студент группы К32422

Плеханова Д.Д.

Санкт-Петербург 2022/2023

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, часть 2 и 3).
2. Составить 3 запроса на модификацию данных (INSERT, UPDATE, DELETE) с использованием подзапросов.
3. Изучить графическое представление запросов и посмотреть 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 работы:

Предметная область – отель (вариант 1)

Наименование БД – Hotel

Наименование схемы - hotel

Схемы логической модели базы данных

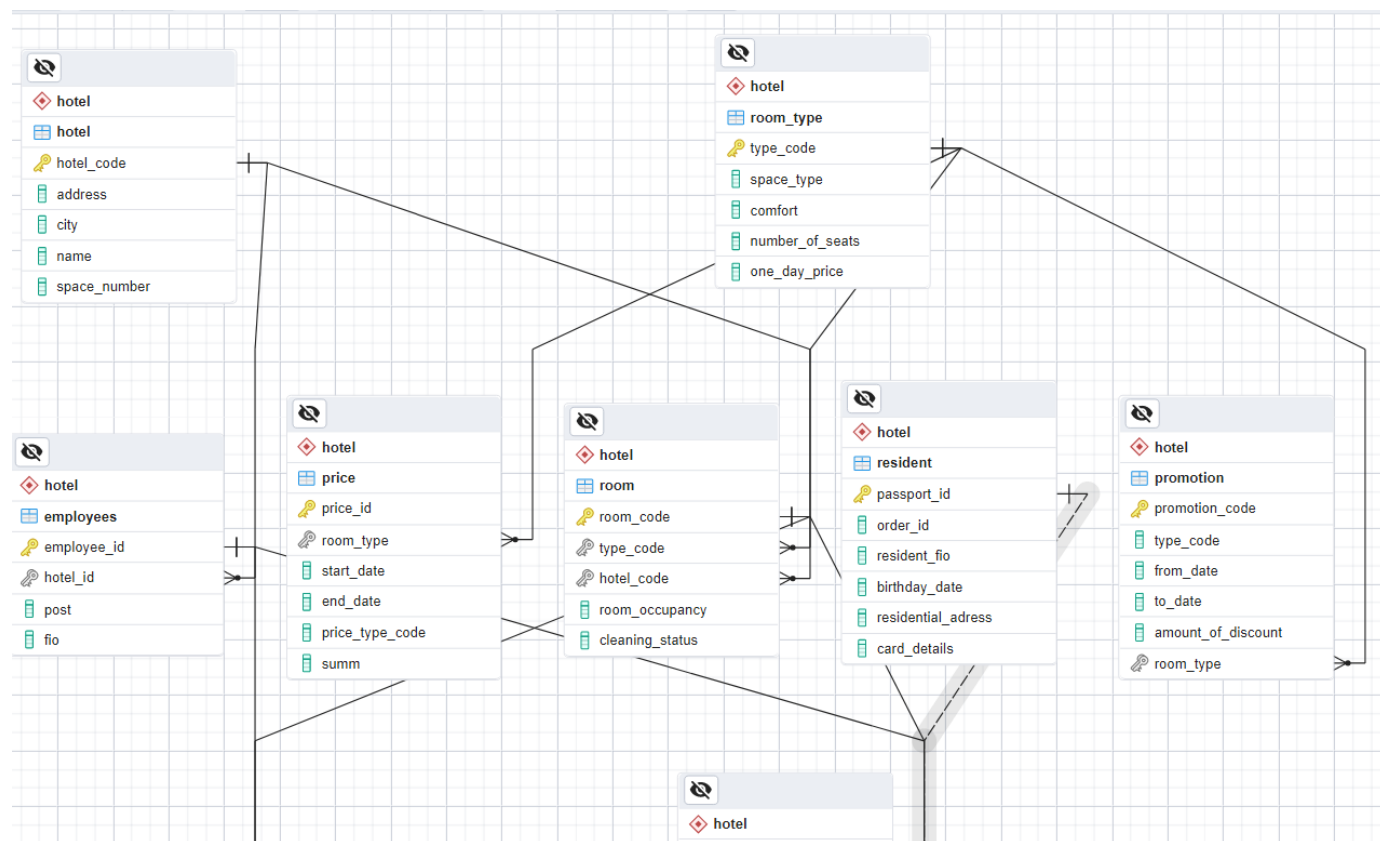


Рисунок 1 – ERD базы данных 1

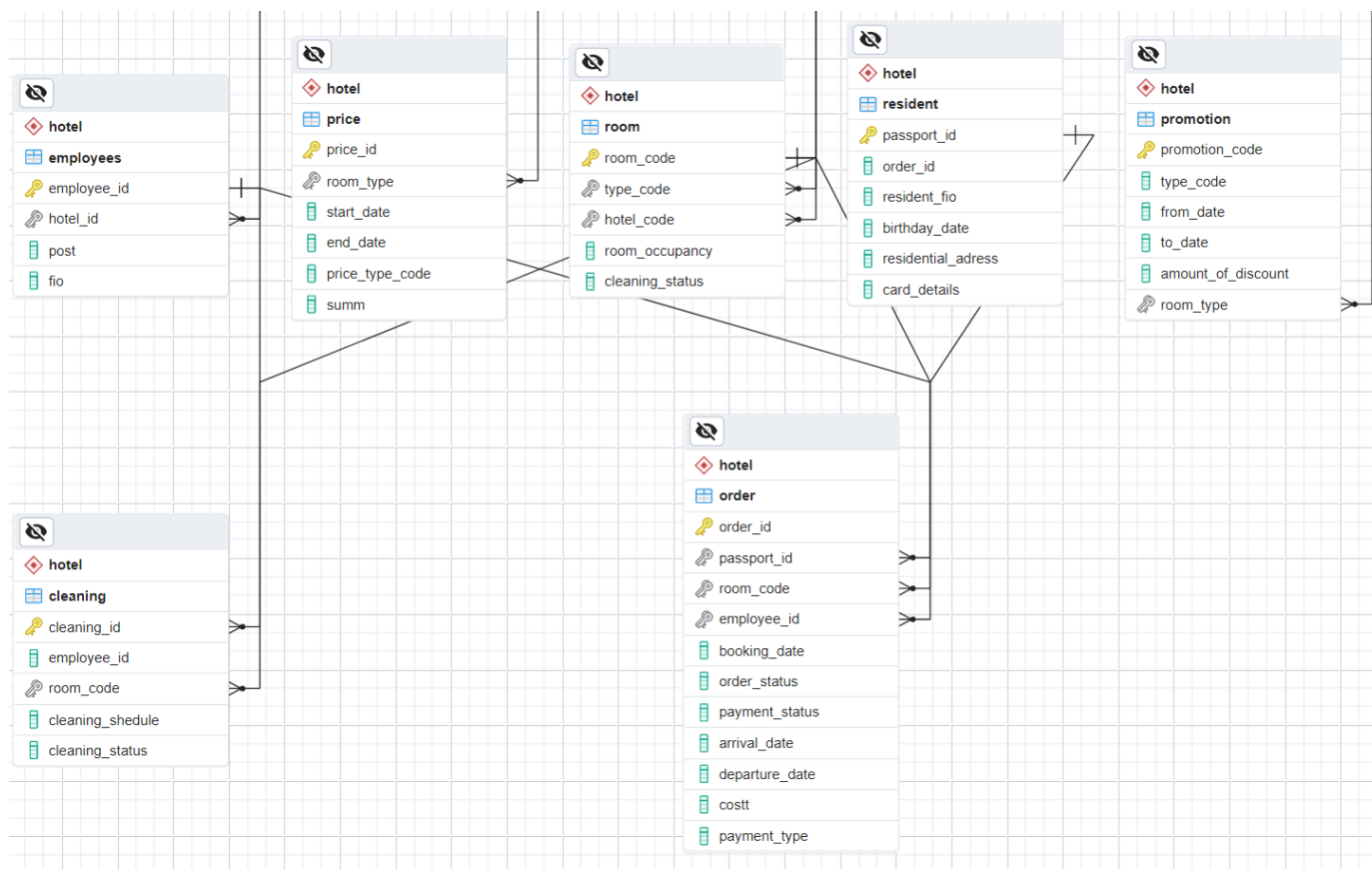


Рисунок 2 – ERD базы данных 2

Запросы на выборку

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

Запрос:

Query









Query History


```
1 SELECT hotel_code
2 FROM hotel.hotel
3     JOIN hotel.room USING(hotel_code)
4     JOIN hotel.room_type USING(type_code)
5     JOIN hotel.price ON room_type.type_code = price.room_type
6 GROUP BY hotel_code
7 HAVING COUNT(room_code) = 2
8     AND SUM(summ) < 200000
9 ORDER BY SUM(summ) DESC;
```

Data Output

Messages

Notifications



	hotel_code [PK] integer 
1	1

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

Запрос:

Query Query History

```
1 SELECT hotel.order.order_id
2 FROM hotel.resident
3     JOIN hotel.order USING(passport_id)
4     JOIN hotel.room USING(room_code)
5     JOIN hotel.hotel USING(hotel_code)
6 WHERE departure_date::date > (NOW()::date - INTERVAL '1 day' * 14)
7 GROUP BY passport_id, hotel.order.order_id;
```

Data Output Messages Notifications



	order_id [PK] integer
1	1

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

Запрос:

Query

Query History

1

SELECT name, ROUND(SUM(summ) / 12) "Цена за месяц"

2

FROM hotel.hotel

3

JOIN hotel.room USING(hotel_code)

4

JOIN hotel.room_type rt USING(type_code)

5

JOIN hotel.price p ON rt.type_code = p.room_type

6

JOIN hotel."order" USING(room_code)

7

WHERE booking_date >= (NOW() - INTERVAL '1 month')

8

GROUP BY hotel_code;

Data Output

Messages

Notifications

≡+

▼

	name character varying	Цена за месяц double precision
1	Fake Hostel Дыбенко	166

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

Запрос:

Query

Query History

1

SELECT name, room_code

2

FROM hotel.hotel

3

JOIN hotel.room USING(hotel_code)

4

WHERE room_occupancy = 'Свободно' AND name = 'Fake Hostel Дыбенко';

Data Output

Messages

Notifications

≡+

▼

	name character varying	room_code integer
1	Fake Hostel Дыбенко	1
2	Fake Hostel Дыбенко	2

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

Запрос:

Query

Query History

1

SELECT SUM(one_day_price)"Общие потери за день"

2

FROM hotel.room

3

JOIN hotel.room_type USING(type_code)

4

WHERE room_occupancy = 'Свободно'

Data Output

Messages

Notifications

≡+

Общие потери за день

bigint

1

7700

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

Запрос:

Query

Query History

1

2

3

4

5

6

7

8

9

10

11

12

13

14


```
SELECT name
FROM hotel.hotel
JOIN hotel.room USING(hotel_code)
WHERE room_occupancy = 'Свободно'
GROUP BY hotel_code
HAVING COUNT(room_code) = (
    SELECT MAX(cnt) FROM (
        SELECT COUNT(room_code) cnt
        FROM hotel.room
        WHERE room_occupancy = 'Свободно'
        GROUP BY hotel_code
    ) T
);
```


Data Output


Messages


Notifications


≡+


















	name	
1	Fake Hostel Дыбенко	

7. Определить самый популярный тип номеров за последний год.

Запрос:

Query

Query History

1

SELECT type_code, COUNT(*) "Количество заказов"

2

FROM hotel.room_type

3

JOIN hotel.room USING(type_code)

4

JOIN hotel."order" USING(room_code)

5

WHERE booking_date >= NOW()-interval '1 year'

6

GROUP BY type_code

7

ORDER BY 2 DESC

8

LIMIT 1;

Data Output

Messages

Notifications

≡+

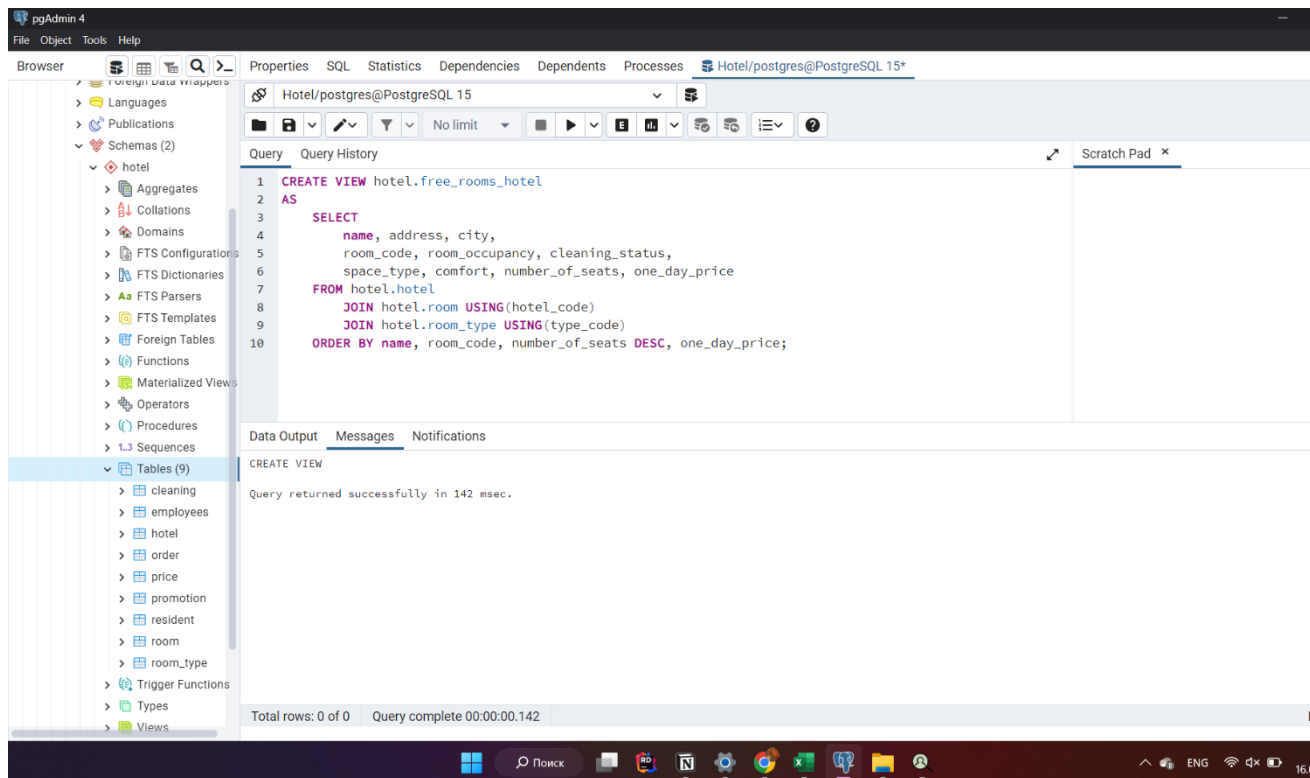
▼

	type_code [PK] integer	Количество заказов bigint
1	1	1

Создание представлений

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

Запрос:



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

Запрос:

The screenshot shows the pgAdmin 4 web interface. On the left, the 'Browser' pane displays a tree structure of database objects, with 'hotel' selected under 'Schemas (2)'. The main pane shows a SQL query editor with the following code:

```
1 CREATE VIEW hotel_income AS
2 (
3     SELECT hotel.hotel_code,
4           hotel.name,
5           SUM(("order".departure_date - "order".arrival_date) * room_type.one_n
6     FROM hotel
7           JOIN room USING (hotel_code)
8           JOIN room_type USING (type_code)
9           JOIN "order" USING (room_code)
10    WHERE booking_date >= CURRENT_DATE - INTERVAL '1 month'
11          AND payment_status = 'оплачен'
12    GROUP BY hotel.hotel_code, hotel.name
13 )
14
```

Below the query editor, the 'Messages' pane shows the execution result: 'CREATE VIEW' and 'Query returned successfully in 246 msec.' The status bar at the bottom indicates 'Total rows: 0 of 0' and 'Query complete 00:00:00.246'.

Сделан подсчет через количество дней умноженное на цену комнаты за день

Запросы на модификацию данных

1. Insert

Запрос:

The screenshot shows the pgAdmin 4 interface. On the left, the 'hotel' schema is expanded, showing various database objects. The 'Tables (10)' folder is selected. The main pane displays the SQL query editor with the following query:

```
1 INSERT INTO hotel.resident (passport_id, order_id, resident_fio, birthday_date,
2 SELECT '2020405406', order_id, 'John Smith', '1990-01-01', '123 Main Street', ''
3 FROM hotel.order
4 WHERE order_id = 1;
```

The query is executed successfully, returning 1 row in 159 msec. The status bar at the bottom indicates 'Query returned successfully in 159 msec'.

Вывод:

The screenshot shows the pgAdmin 4 interface. The 'resident' table is selected in the left pane. The main pane displays the SQL query editor with the following query:

```
1 SELECT * FROM hotel.resident
2 ORDER BY passport_id ASC
```

The query is executed successfully, returning 5 rows in 364 msec. The status bar at the bottom indicates 'Successfully run. Total query runtime: 364 msec. 5 rows affected'.

	passport_id [PK] bigint	order_id integer	resident_fio character varying (50)	birthday_date date	residentialAddress character varying	card_details character varying
1	2020405406	1	John Smith	1990-01-01	123 Main Street	1234567890123456
2	4017899234	2	Иванов Иван Иванович	2000-09-09	Гастелло 17	2222 3333 4444 5555 11/34 111
3	4018438576	1	Абдурахман ибн Хатаб	2003-06-16	Московский пр д 45	2200 1519 9114 7191 11/29 081
4	4018748562	1	Смирнова Мария Ивановна	2000-01-01	Московский пр д 45	2200 1519 9114 7191 11/29 081

2. Update

Запрос:

The screenshot shows the pgAdmin 4 interface. In the left sidebar, the 'hotel' schema is expanded, and the 'resident' table is selected. The main pane displays the following SQL query:

```
1 UPDATE hotel.resident
2 SET residential_address = 'новый_адрес'
3 WHERE passport_id IN (
4     SELECT passport_id
5     FROM hotel.order
6     WHERE order_status = 'невыполнен'
7 );
```

The 'Messages' tab at the bottom shows the execution result:

```
UPDATE 1
Query returned successfully in 136 msec.
```

The status bar at the bottom indicates 'Total rows: 0 of 0' and 'Query complete 00:00:00.136'.

Вывод:

The screenshot shows the pgAdmin 4 interface with the 'resident' table selected. The main pane displays the following SQL query:

```
1 SELECT * FROM hotel.resident
2 ORDER BY passport_id ASC
```

The 'Data Output' tab at the bottom shows the query results in a table with 7 columns: passport_id [PK] bigint, order_id integer, resident_fio character varying (50), birthday_date date, residential_address character varying, and card_details character varying. The results are as follows:

	passport_id [PK] bigint	order_id integer	resident_fio character varying (50)	birthday_date date	residential_address character varying	card_details character varying
1	2020405406	1	John Smith	1990-01-01	123 Main Street	1234567890123456
2	4017899234	2	Иванов Иван Иванович	2000-09-09	Гастелло 17	2222 3333 4444 5555 11/34 111
3	4018438576	1	Абдурахман ибн Хатаб	2003-06-16	новый_адрес	2200 1519 9114 7191 11/29 081
4	4018748562	1	Смирнова Мария Ивановна	2000-01-01	Московский пр д 45	2200 1519 9114 7191 11/29 081

The status bar at the bottom indicates 'Total rows: 5 of 5' and 'Query complete 00:00:01.262'.

3. Delete

Запрос:

The screenshot displays the pgAdmin 4 web interface. On the left, the 'Browser' pane shows the database structure, with 'Tables (10)' expanded under the 'hotel' schema. The central pane shows a SQL query being executed in the 'Query' tab. The query is a DELETE statement that removes records from the 'hotel.cleaning' table based on room codes selected from the 'hotel.room' table where the cleaning status is 'Убрано' and the hotel code is 1. The 'Messages' tab at the bottom shows the successful execution of the query, returning 0 rows in 175 milliseconds. The status bar at the bottom indicates 'Total rows: 0 of 0' and 'Query complete 00:00:00.175'.

pgAdmin 4

File Object Tools Help

Browser

- Publications
- Schemas (2)
 - hotel
 - Aggregates
 - Collations
 - Domains
 - FTS Configurations
 - FTS Dictionaries
 - FTS Parsers
 - FTS Templates
 - Foreign Tables
 - Functions
 - Materialized Views
 - Operators
 - Procedures
 - Sequences
 - Tables (10)**
 - cleaning
 - employees
 - hotel
 - order
 - order_log
 - price
 - promotion
 - resident
 - room
 - room_type
 - Trigger Functions
 - Types
 - Views
 - public

Hotel/postgres@PostgreSQL 15

No limit

Query Query History

```
1
2 DELETE FROM hotel.cleaning
3 WHERE room_code IN (
4     SELECT room_code
5     FROM hotel.room
6     WHERE cleaning_status = 'Убрано' AND hotel_code = 1
7 );
```

Scratch Pad

Data Output Messages Notifications

DELETE 0

Query returned successfully in 175 msec.

Total rows: 0 of 0 Query complete 00:00:00.175 Ln 6, Col 56

21:08 28.06.2023

Вывод:

Data Output Messages Notifications						
<div><div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div></div></div>						
	cleaning_id [PK] integer	employee_id integer	room_code integer	cleaning_shedule character varying	cleaning_status character varying (20)	
1	2	1	1	Еженедельное	Выполнена	
2	3	2	2	Еженедельное	Не выполнена	

Сравнение запросов с использованием индексирования

1. Сравнение простого индекса

Запрос:

Query Query History

1

EXPLAIN ANALYZE SELECT * FROM hotel.resident WHERE resident_fio = 'Иванов Иван Иванович';

Data Output Messages Notifications

QUERY PLAN		
text		
1	Seq Scan on resident (cost=0.00..14.50 rows=2 width=198) (actual time=0.043..0.045 rows=1 loops...	
2	Filter: ((resident_fio)::text = 'Иванов Иван Иванович')::text)	
3	Rows Removed by Filter: 3	
4	Planning Time: 2.752 ms	
5	Execution Time: 0.372 ms	

Создание индекса:

Query Query History

1 **CREATE INDEX** idx_resident_fio **ON** hotel.resident(resident_fio);

Data Output **Messages** Notifications

CREATE INDEX

Query returned successfully in 182 msec.

Время выполнения с индексом:

Query

Query History

1

EXPLAIN ANALYZE SELECT * FROM hotel.resident WHERE resident_fio = 'Иванов Иван Иванович';

Data Output

Messages

Notifications

QUERY PLAN

text

1

Seq Scan on resident (cost=0.00..1.05 rows=1 width=198) (actual time=0.029..0.030 rows=1 loops...

2

Filter: ((resident_fio)::text = 'Иванов Иван Иванович':::text)

3

Rows Removed by Filter: 3

4

Planning Time: 3.489 ms

5

Execution Time: 0.051 ms

2. Сравнение составного индекса

Запрос:

Query Query History

```
1 EXPLAIN ANALYZE SELECT *
2 FROM hotel.employees
3 WHERE post = 'Директор' AND hotel_id = 2;
4
```

Data Output Messages Notifications



	QUERY PLAN	
	text	
1	Seq Scan on employees (cost=0.00..15.25 rows=1 width=204) (actual time=0.020..0.021 rows=1 loops...	
2	Filter: (((post)::text = 'Директор'::text) AND (hotel_id = 2))	
3	Rows Removed by Filter: 2	
4	Planning Time: 11.844 ms	
5	Execution Time: 0.042 ms	

Создание индекса:

Query Query History

```
1 EXPLAIN ANALYZE SELECT *
2 FROM hotel.employees
3 WHERE post = 'Директор' AND hotel_id = 2;
4 CREATE INDEX idx_employees_post_hotelid ON hotel.employees(post, hotel_id);
```

Data Output Messages Notifications

CREATE INDEX

Query returned successfully in 437 msec.

Время выполнения с индексом:

Query Query History

```
1 EXPLAIN ANALYZE SELECT *
2 FROM hotel.employees
3 WHERE post = 'Директор' AND hotel_id = 2;
```

Data Output Messages Notifications



QUERY PLAN

text



1	Seq Scan on employees (cost=0.00..1.04 rows=1 width=204) (actual time=0.029..0.029 rows=1 loops...
2	Filter: (((post)::text = 'Директор'::text) AND (hotel_id = 2))
3	Rows Removed by Filter: 2
4	Planning Time: 2.220 ms
5	Execution Time: 0.056 ms

Заключение:

В ходе выполнения данной лабораторной работе были реализованы запросы на выборку данных и представления к базе данных на PostgreSQL согласно индивидуальному заданию, изучены запросы на модификацию данных и созданы индексы для различных запросов.