

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

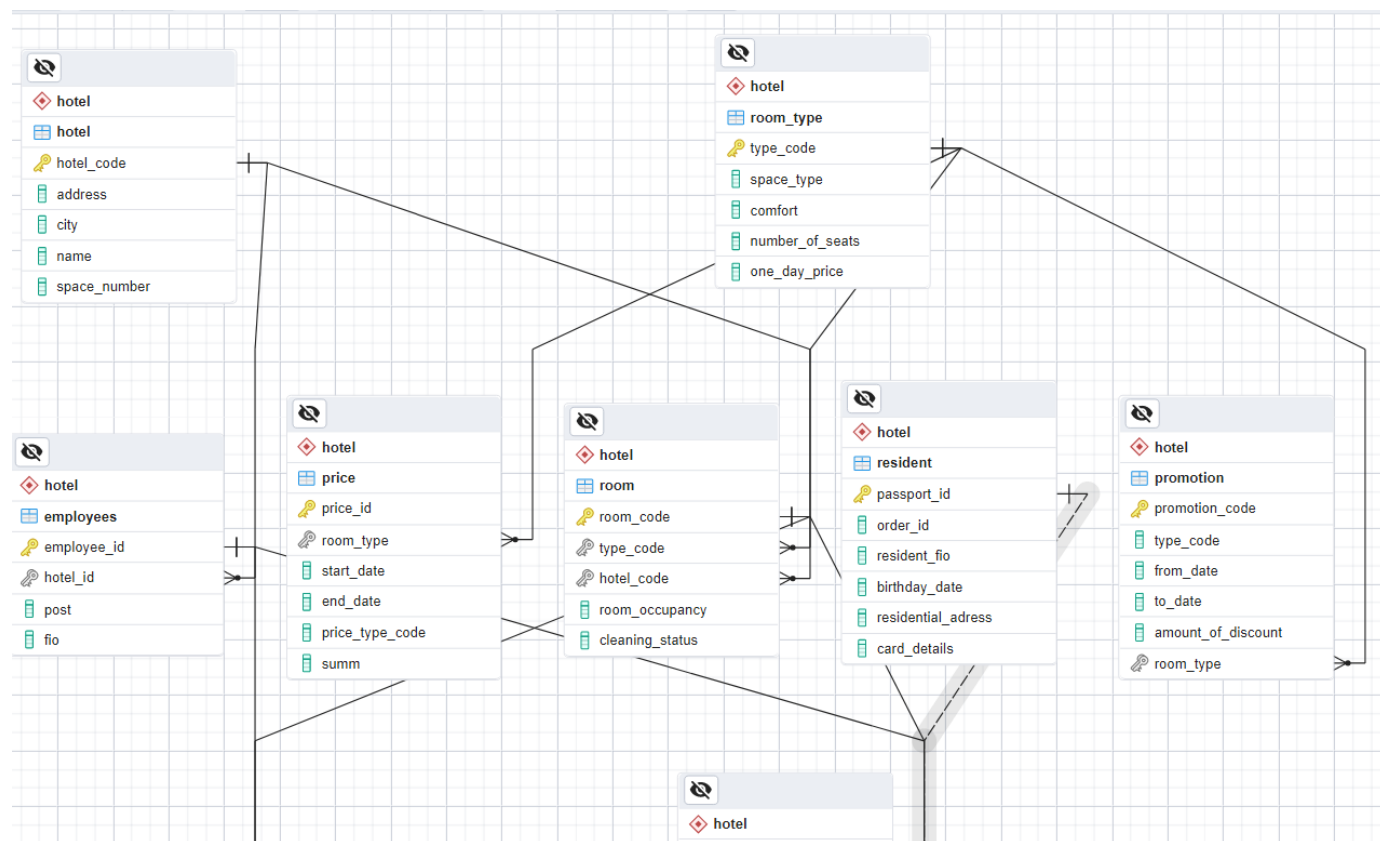


Рисунок 1 – ERD базы данных 1

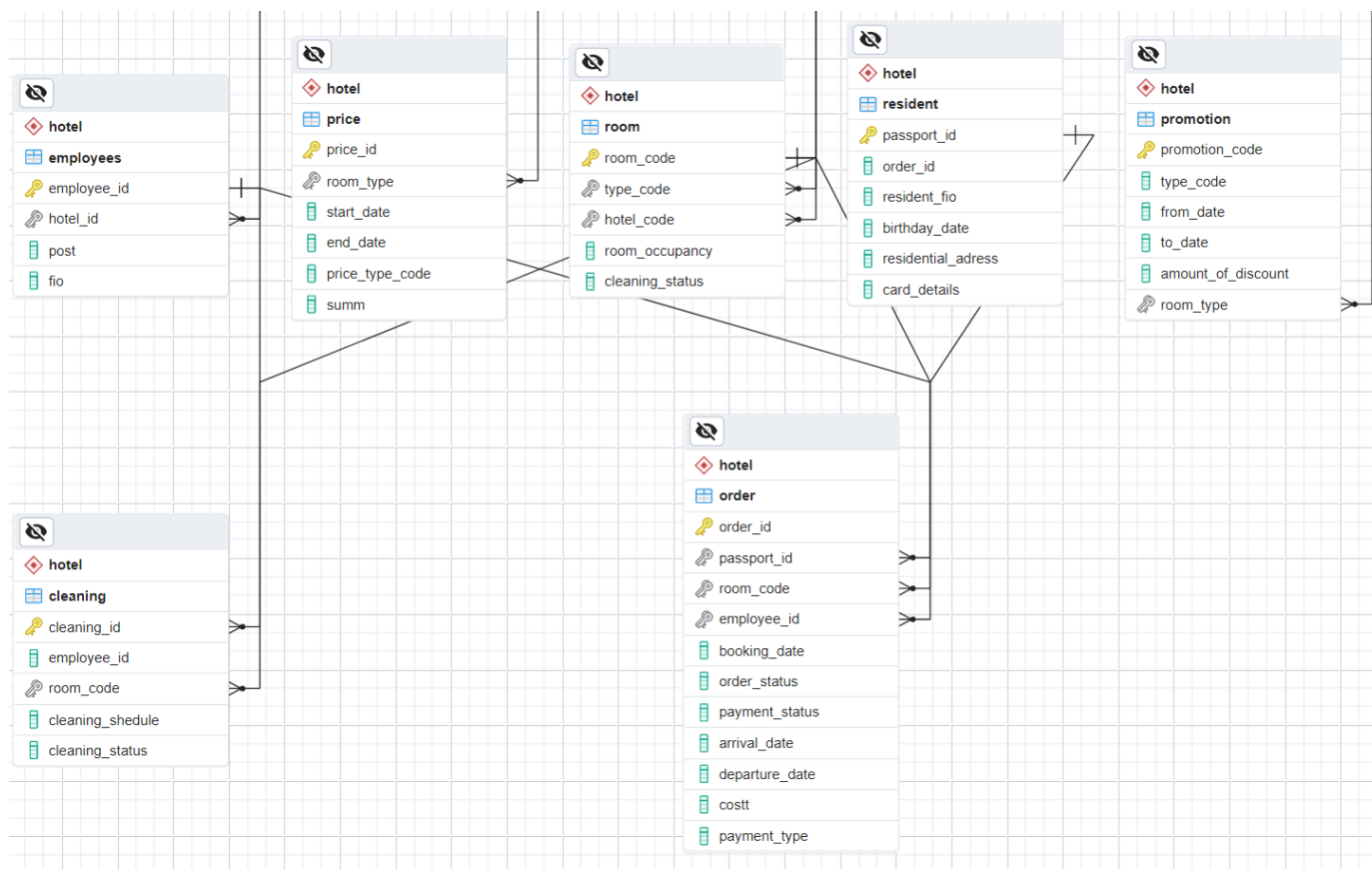


Рисунок 2 – ERD базы данных 2

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 SELECT name
2 FROM hotel.hotel
3     JOIN hotel.room USING(hotel_code)
4 WHERE room_occupancy = 'Свободно'
5 GROUP BY hotel_code
6 HAVING COUNT(room_code) = (
7     SELECT MAX(cnt) FROM (
8         SELECT COUNT(room_code) cnt
9         FROM hotel.room
10        WHERE room_occupancy = 'Свободно'
11        GROUP BY hotel_code
12    ) T
13 );
14
```

Data Output

Messages

Notifications



	name	
	character varying	
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

≡+

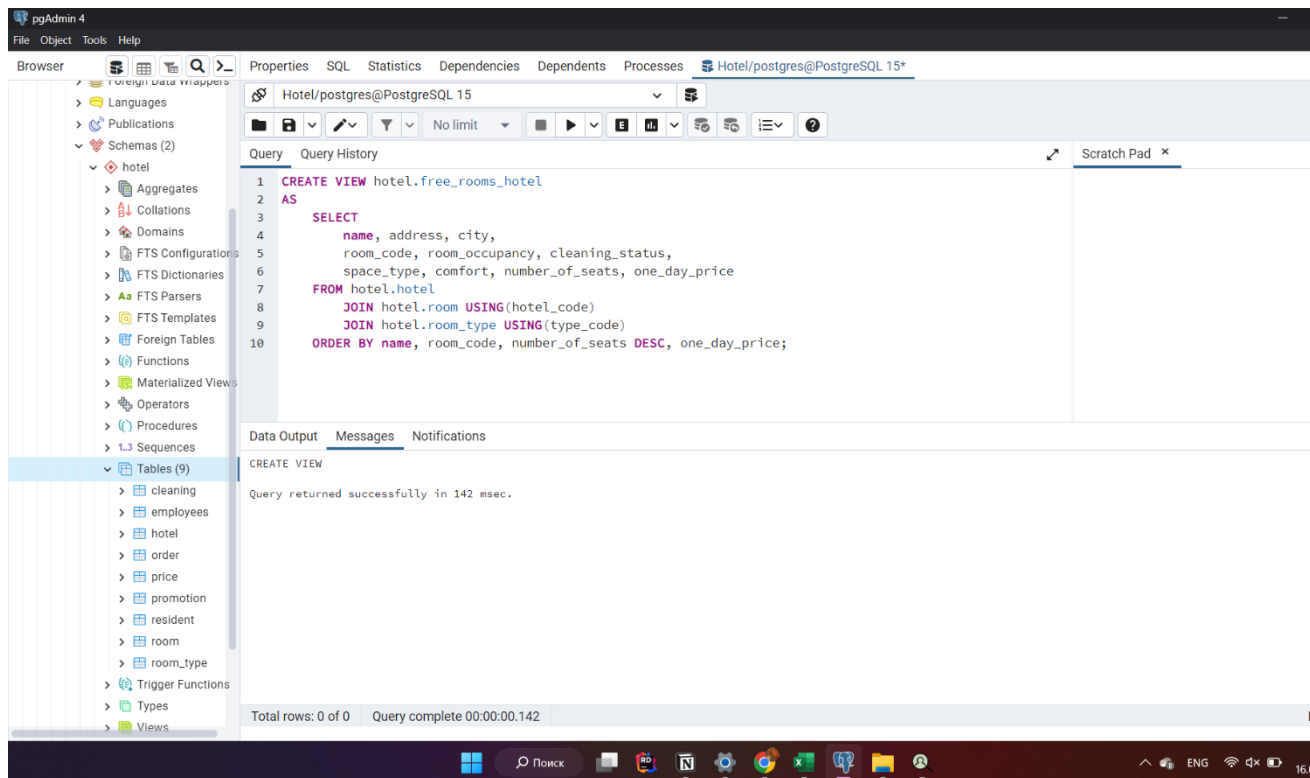
▼

	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:

Запросы на модификацию данных

1. Insert

Запрос:

Query

Query History

1

2

3

```
INSERT INTO hotel.price(  
    price_id, room_type, start_date, end_date, price_type_code, summ)  
VALUES (5, 2, '2023-07-07','2023-08-08' , 'GF445G', 7700);
```

Data Output

Messages

Notifications

INSERT 0 1

Query returned successfully in 118 msec.

Вывод:

	price_id [PK] integer	room_type integer	start_date date	end_date date	price_type_code character varying (40)	summ integer
1	1	1	2023-01-01	2023-02-02	F23GH	2000
2	2	2	2023-02-03	2023-03-01	F23BH	3000
3	3	3	2023-01-01	2023-02-02	F23TTH	2500
4	5	2	2023-07-07	2023-08-08	GF445G	7700

2. Update

Запрос:

Query

Query History

1

UPDATE hotel.hotel

2

SET address='ул Гагарина д.45', city='Москва', space_number=100

3

WHERE hotel_code=1;

Data Output

Messages

Notifications

UPDATE 1

Query returned successfully in 116 msec.

Вывод:

Data Output

Messages

Notifications

≡+

📄

▼

📋

🗑️

🗄️

⬇️

📈

	hotel_code [PK] integer	address character varying	city character varying	name character varying	space_num integer
1	1	ул Гагарина д.45	Москва	Fake Hostel Дыбенко	
2	2	ул. Плеханова д. 14	Воронеж	Fake Hostel Воронеж	
3	3	ул. Образцовая д. 7 к. 1 с. 3	Мурино	Fake Hostel Мурино	

3. Delete

Запрос:

Query Query History

```
1 DELETE FROM hotel.cleaning
2 WHERE cleaning_id=1;
```

Data Output Messages Notifications

DELETE 1

Query returned successfully in 106 msec.

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.