Министерство науки и высшего образования Российской Федерации
федеральное государственное автономное образовательное
учреждение высшего образования
“Национальный исследовательский университет ИТМО”

Факультет инфокоммуникационных технологий

ЛАБОРАТОРНАЯ РАБОТА №2

ЗАПРОСЫ НА ВЫБОРКУ И МОДИФИКАЦИЮ ДАННЫХ, ПРЕДСТАВЛЕНИЯ И ИНДЕКСЫ В POSTGRESQL

по дисциплине:
«Проектирование и реализация баз данных»

Выполнил студент:
Старовойтова Елизавета Анатольевна
Группа №K32402
Преподаватель:
Говорова Марина Михайловна

Санкт-Петербург
2023

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

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.

Индивидуальное задание (вариант):

Вариант 1. БД «Отель»

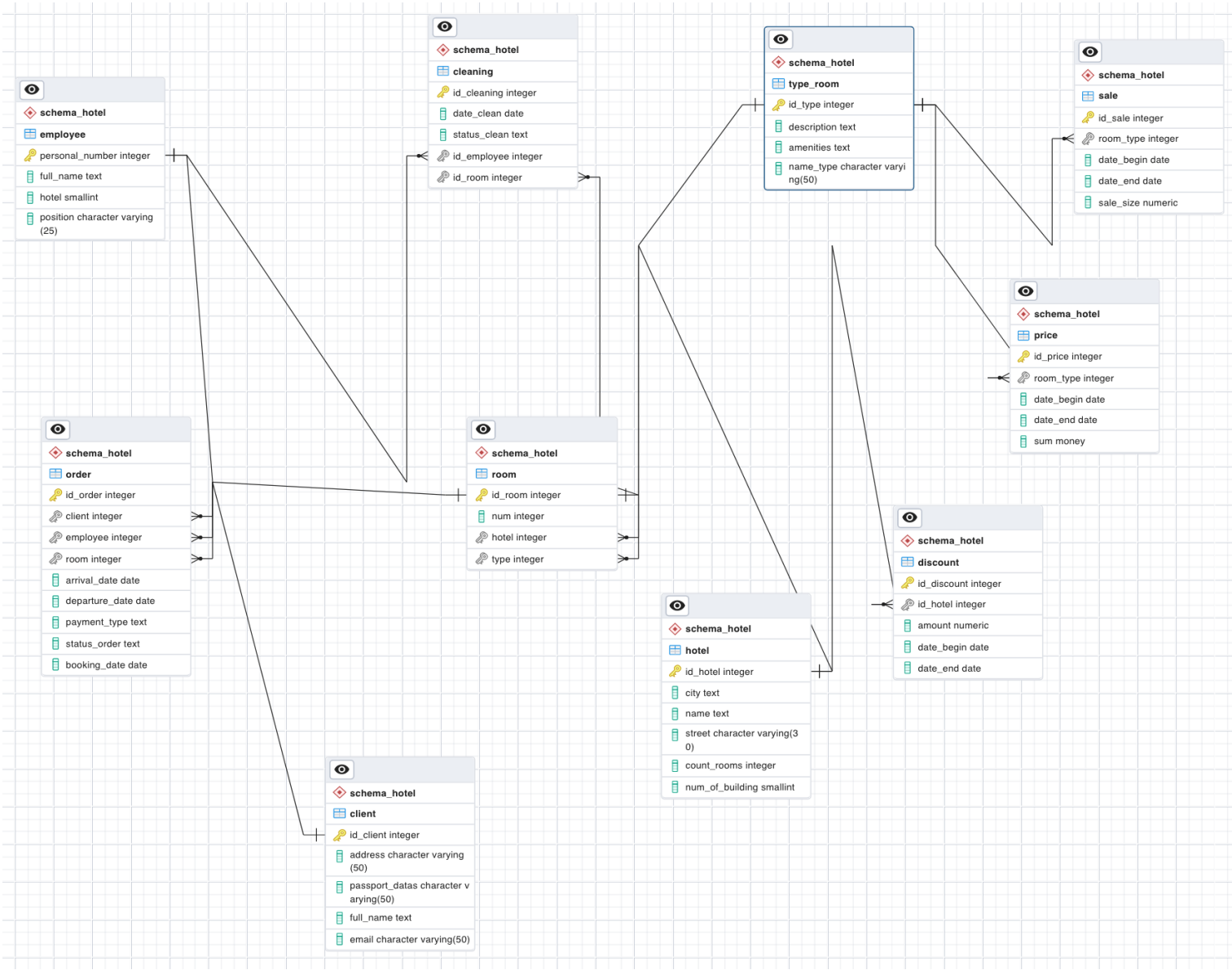
Описание предметной области: Отели сети находятся в разных городах. Цены на номера одного типа во всех отелях одинаковы и зависят от типа номера и количества мест. Номер может быть забронирован, занят или свободен. При заезде в отель постояльцы проходят регистрацию. Информация о регистрации постояльцев отеля (выехавших из отеля) хранится в течение года и 1 января удаляется в архив.

Номера ежедневно убираются горничными, для чего составляется график уборки номеров. Ежедневно каждому номеру присваивается статус “убран”, “не убран”. Цены на номера могут меняться.

БД должна содержать следующий минимальный набор сведений: Адрес отеля. Название отеля. Номер комнаты. Тип комнаты. Количество мест. Удобства. Цена комнаты за сутки проживания. Имя постояльца. Фамилия постояльца. Отчество постояльца. Адрес постоянного проживания. Дата заезда. Дата отъезда. График уборки номеров.

Дополнить исходные данные информацией: по бронированию комнаты; по сотруднику, который регистрирует постояльца в отеле в день заезда; по оплате проживания; по составу удобств в комнате; по акциям, доступным при бронировании (скидки).

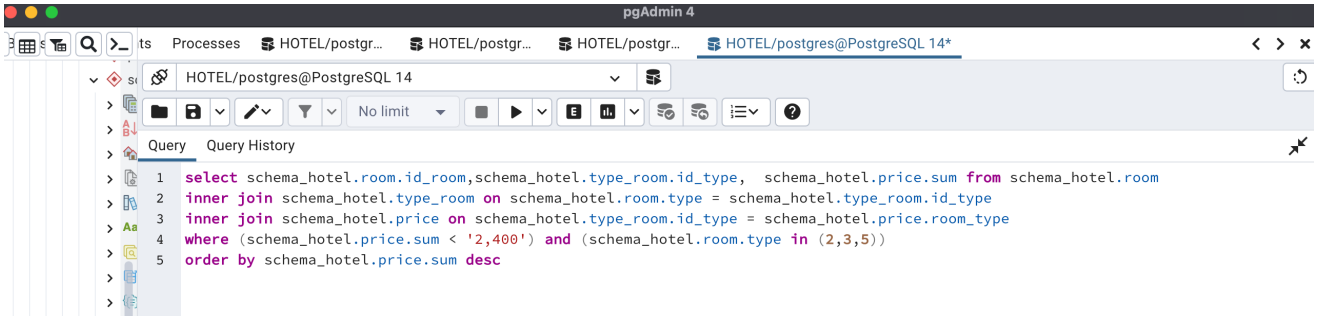
Схема базы данных:



Выполнение:

Запросы к базе данных:

1. Составить список всех 2-х местных номеров отелей, с ценой менее 200 т. р. ($2450), упорядочив данные в порядке уменьшения стоимости.



Data Output Messages Notifications

	id_room integer	id_type integer	sum money
1	1	2	$150.00
2	2	2	$150.00
3	5	2	$150.00
4	6	2	$150.00
5	11	2	$150.00
6	12	2	$150.00
7	13	3	$30.00
8	14	3	$30.00
9	19	3	$30.00
10	20	3	$30.00
11	24	3	$30.00

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

```
select schema_hotel.order.id_order, schema_hotel.order.client, schema_hotel.order.employee, schema_hotel.order.room, schema_hotel.order.arrival_date, schema_hotel.order.departure_date from schema_hotel.order
inner join schema_hotel.client on schema_hotel.client.id_client = schema_hotel.order.client
where schema_hotel.order.departure_date between now()- interval '2 week' and now()
```

Data Output Messages Notifications

	id_order [PK] integer	client integer	employee integer	room integer	arrival_date date	departure_date date
1	1	1	3000	10	2023-04-14	2023-04-16
2	3	2	3001	6	2023-04-14	2023-04-16
3	4	3	3002	3	2023-04-13	2023-04-14
4	5	4	3000	16	2023-04-22	2023-04-23

3. Чему равен общ. сут. доход каждого отеля за последний месяц?

Query Query History

```
select schema_hotel.hotel.id_hotel, schema_hotel.hotel.name,
sum(schema_hotel.price.sum * extract(day from AGE(departure_date, arrival_date)))/30
as daily_income from schema_hotel.order
inner join schema_hotel.room on schema_hotel.room.id_room = schema_hotel.order.room
inner join schema_hotel.hotel on schema_hotel.room.hotel = schema_hotel.hotel.id_hotel
inner join schema_hotel.price on schema_hotel.room.type = schema_hotel.price.room_type
where schema_hotel.order.arrival_date >= '2023-03-28' and schema_hotel.order.departure_date <= '2023-04-28'
group by schema_hotel.hotel.id_hotel, schema_hotel.hotel.name
```

Data Output Messages Notifications

	id_hotel [PK] integer	name text	daily_income money
1	2	Rixos MINSK	$40.00
2	1	Rixos SPB	$86.66
3	3	Rixos EKB	$65.66

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

uery Query History

```
select schema_hotel.room.id_room, schema_hotel.room.num, schema_hotel.room.type from schema_hotel.room
left join (select schema_hotel.order.room from schema_hotel.order where arrival_date <= current_date
           and departure_date >= current_date
          ) as o on schema_hotel.room.id_room = o.room
where schema_hotel.room.hotel = 2 and o.room is null
```

ata Output Messages Notifications

	id_room [PK] integer	num integer	type integer
	6	20	2
	7	30	1
	8	40	1
	17	25	5
	18	24	5
	19	23	3
	20	22	3

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

Query Query History

```
select sum(schema_hotel.price.sum) from schema_hotel.room
join schema_hotel.type_room on schema_hotel.type_room.id_type = schema_hotel.room.type
join schema_hotel.price on schema_hotel.price.room_type = schema_hotel.type_room.id_type
left join (select schema_hotel.order.room from schema_hotel.order where arrival_date <= current_date
           and departure_date >= current_date
          ) as o on schema_hotel.room.id_room = o.room
where o.room is null
```

Data Output Messages Notifications

	sum money
1	$13,770.00

6. Определить, в каком отеле имеется наибольшее кол-во незанятых номеров на текущие сутки.

Query Query History

```
select schema_hotel.hotel.name, count(schema_hotel.room.id_room) as available_rooms
from schema_hotel.hotel
join schema_hotel.room on schema_hotel.room.hotel = schema_hotel.hotel.id_hotel
left join (select schema_hotel.order.room from schema_hotel.order where arrival_date <= current_date
           and departure_date >= current_date
          ) as o on schema_hotel.room.id_room = o.room
where o.room is null
group by schema_hotel.hotel.id_hotel
order by available_rooms desc
limit 1
```

Data Output Messages Notifications

	name text	available_rooms bigint
1	Rixos MINSK	7

7. Определить самый популярный тип номеров за последний год.

Query Query History

```
select schema_hotel.type_room.name_type, count(*) as num_bookings
from schema_hotel.order
join schema_hotel.room on schema_hotel.order.room = schema_hotel.room.id_room
join schema_hotel.type_room on schema_hotel.room.type = schema_hotel.type_room.id_type
where schema_hotel.order.departure_date >= DATE_TRUNC('year', current_date) - INTERVAL '1 year'
group by schema_hotel.type_room.id_type
order by num_bookings desc
limit 1;
```

Data Output Messages Notifications

	name_type character varying (50)	num_bookings bigint
1	double comfort	3

Представления:

1. Для турагентов (поиск свободных номеров в отелях)

```
CREATE VIEW available_rooms AS
SELECT schema_hotel.room.id_room AS room_id, schema_hotel.hotel.name AS hotel_name, schema_hotel.type_room.name_type AS room_type, schema_hotel.room.num AS room_number
FROM schema_hotel.room
JOIN schema_hotel.hotel ON schema_hotel.room.hotel = schema_hotel.hotel.id_hotel
JOIN schema_hotel.type_room ON schema_hotel.room.type = schema_hotel.type_room.id_type
WHERE NOT EXISTS (
    SELECT 1 FROM schema_hotel.order
    WHERE schema_hotel.order.room = schema_hotel.room.id_room AND schema_hotel.order.arrival_date <= CURRENT_DATE AND schema_hotel.order.departure_date >= CURRENT_DATE
)
SELECT * FROM available_rooms
```

	room_id integer	hotel_name text	room_type character varying (50)	room_number integer
1	1	Rixos EKB	double comfort	10
2	3	Rixos EKB	family standart	30
3	4	Rixos EKB	family standart	40
4	6	Rixos MINSK	double comfort	20
5	7	Rixos MINSK	family standart	30
6	8	Rixos MINSK	family standart	40
7	9	Rixos SPB	family standart	40
8	10	Rixos SPB	family standart	30
9	11	Rixos SPB	double comfort	20
10	12	Rixos SPB	double comfort	10
11	13	Rixos SPB	double standart	25
12	14	Rixos SPB	double standart	35
13	15	Rixos SPB	family comfort	45
14	17	Rixos MINSK	double suite	25
15	18	Rixos MINSK	double suite	24
16	19	Rixos MINSK	double standart	23
17	20	Rixos MINSK	double standart	22
18	21	Rixos EKB	double suite	21
19	22	Rixos EKB	double suite	24
20	23	Rixos EKB	double suite	22

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

Query Query History

```
1  CREATE VIEW hotel_income AS
2  SELECT schema_hotel.hotel.id_hotel, schema_hotel.hotel.name, SUM(schema_hotel.price.sum)
3  FROM schema_hotel.hotel
4  JOIN schema_hotel.room ON schema_hotel.hotel.id_hotel = schema_hotel.room.hotel
5  JOIN schema_hotel.type_room ON schema_hotel.type_room.id_type = schema_hotel.room.type
6  JOIN schema_hotel.price ON schema_hotel.type_room.id_type = schema_hotel.price.room_type
7  LEFT JOIN schema_hotel.order o ON schema_hotel.room.id_room = o.room AND o.departure_date >= DATE_TRUNC('month', CURRENT_DATE)
8  AND o.arrival_date < DATE_TRUNC('month', CURRENT_DATE + INTERVAL '1 month')
9  WHERE o.id_order IS NULL
10 GROUP BY schema_hotel.hotel.id_hotel, schema_hotel.hotel.name
11 SELECT * FROM hotel_income
```

Data Output Messages Notifications

	id_hotel integer	name text	sum money
1	1	Rixos SPB	$660.00
2	2	Rixos MINSK	$5,160.00
3	3	Rixos EKB	$7,700.00

Запросы на модификацию данных:

1. Создать бронь для определенного отеля, которая включает один из номеров с максимальной стоимостью.

```
INSERT INTO schema_hotel.order (client, employee, room, arrival_date, departure_date, payment_type, status_order, booking_date)
VALUES (1, 3002,
    (SELECT schema_hotel.room.id_room
     FROM schema_hotel.room
     INNER JOIN schema_hotel.type_room ON schema_hotel.type_room.id_type = schema_hotel.room.type
     INNER JOIN schema_hotel.price ON schema_hotel.price.room_type = schema_hotel.type_room.id_type
     WHERE schema_hotel.room.hotel = 3
     AND schema_hotel.room.id_room NOT IN
         (SELECT schema_hotel.order.room
          FROM schema_hotel.order
          WHERE (schema_hotel.order.arrival_date <= '2023-05-10' AND schema_hotel.order.departure_date >= '2023-04-24')
         )
     AND schema_hotel.price.sum =
         (SELECT MAX(schema_hotel.price.sum)
          FROM schema_hotel.room
          INNER JOIN schema_hotel.type_room ON schema_hotel.type_room.id_type = schema_hotel.room.type
          INNER JOIN schema_hotel.price ON schema_hotel.price.room_type = schema_hotel.type_room.id_type
          WHERE schema_hotel.room.hotel = 3
          AND schema_hotel.room.id_room NOT IN
              (SELECT schema_hotel.order.room
               FROM schema_hotel.order
               WHERE (schema_hotel.order.arrival_date <= '2023-05-10' AND schema_hotel.order.departure_date >= '2023-04-24')
              )
         )
     ORDER BY schema_hotel.price.sum DESC
     LIMIT 1
    ),
    '2023-04-24', '2023-05-10', 'mastercard', 'approved', '2023-04-23');
```

	id_order [PK] integer	client integer	employee integer	room integer	arrival_date date	departure_date date	payment_type text	status_order text	booking_date date
1	1	1	3000	10	2023-04-14	2023-04-16	visa	approved	2023-03-20
2	3	2	3001	6	2023-04-14	2023-04-16	mir pay	approved	2023-03-29
3	4	3	3002	3	2023-04-13	2023-04-14	mastercard	approved	2023-03-24
4	5	4	3000	16	2023-04-22	2023-04-23	visa	approved	2023-04-21
5	6	5	3001	5	2023-04-20	2023-04-26	visa	approved	2023-04-19
6	7	6	3002	24	2023-04-15	2023-04-24	visa	approved	2023-04-12
7	8	7	3002	2	2023-04-15	2023-04-26	visa	approved	2023-04-12

	id_order [PK] integer	client integer	employee integer	room integer	arrival_date date	departure_date date	payment_type text	status_order text	booking_date date
1	1	1	3000	10	2023-04-14	2023-04-16	visa	approved	2023-03-20
2	3	2	3001	6	2023-04-14	2023-04-16	mir pay	approved	2023-03-29
3	4	3	3002	3	2023-04-13	2023-04-14	mastercard	approved	2023-03-24
4	5	4	3000	16	2023-04-22	2023-04-23	visa	approved	2023-04-21
5	6	5	3001	5	2023-04-20	2023-04-26	visa	approved	2023-04-19
6	7	6	3002	24	2023-04-15	2023-04-24	visa	approved	2023-04-12
7	8	7	3002	2	2023-04-15	2023-04-26	visa	approved	2023-04-12
8	10	1	3002	21	2023-04-24	2023-05-10	mastercard	approved	2023-04-23

2. Изменить прошлую бронь так, чтобы клиентом был человек, который до этого не совершал заказов в отелях сети.

```sql
UPDATE schema_hotel.order
SET client = (
    SELECT schema_hotel.client.id_client
    FROM schema_hotel.client
    LEFT JOIN schema_hotel.order ON schema_hotel.client.id_client = schema_hotel.order.client
    WHERE schema_hotel.order.client IS NULL
    ORDER BY schema_hotel.client.id_client ASC
    LIMIT 1
)
WHERE id_order = 10;
```

	id_order [PK] integer	client integer	employee integer	room integer	arrival_date date	departure_date date	payment_type text	status_order text	booking_date date
1	1	1	3000	10	2023-04-14	2023-04-16	visa	approved	2023-03-20
2	3	2	3001	6	2023-04-14	2023-04-16	mir pay	approved	2023-03-29
3	4	3	3002	3	2023-04-13	2023-04-14	mastercard	approved	2023-03-24
4	5	4	3000	16	2023-04-22	2023-04-23	visa	approved	2023-04-21
5	6	5	3001	5	2023-04-20	2023-04-26	visa	approved	2023-04-19
6	7	6	3002	24	2023-04-15	2023-04-24	visa	approved	2023-04-12
7	8	7	3002	2	2023-04-15	2023-04-26	visa	approved	2023-04-12
8	10	1	3002	21	2023-04-24	2023-05-10	mastercard	approved	2023-04-23

	id_order [PK] integer	client integer	employee integer	room integer	arrival_date date	departure_date date	payment_type text	status_order text	booking_date date
1	1	1	3000	10	2023-04-14	2023-04-16	visa	approved	2023-03-20
2	3	2	3001	6	2023-04-14	2023-04-16	mir pay	approved	2023-03-29
3	4	3	3002	3	2023-04-13	2023-04-14	mastercard	approved	2023-03-24
4	5	4	3000	16	2023-04-22	2023-04-23	visa	approved	2023-04-21
5	6	5	3001	5	2023-04-20	2023-04-26	visa	approved	2023-04-19
6	7	6	3002	24	2023-04-15	2023-04-24	visa	approved	2023-04-12
7	8	7	3002	2	2023-04-15	2023-04-26	visa	approved	2023-04-12
8	10	8	3002	21	2023-04-24	2023-05-10	mastercard	approved	2023-04-23

7	7	Kosareva 8, Spb	4892 384187	Ivanov Ivan	ivanov@yandex.ru
8	8	Pushkina 8, Minsk	HB3456679	Ivanova Katya Olegovna	ivanova@yandex.ru

3. Удалить заказы со статусом “отменен” прошлого года.

```sql
DELETE FROM schema_hotel.order
WHERE id_order IN (
    SELECT id_order FROM schema_hotel.order
    WHERE status_order = 'cancelled' AND EXTRACT(YEAR FROM booking_date) = 2022
);
```

	id_order [PK] integer	client integer	employee integer	room integer	arrival_date date	departure_date date	payment_type text	status_order text	booking_date date
1	1	1	3000	10	2023-04-14	2023-04-16	visa	approved	2023-03-20
2	3	2	3001	6	2023-04-14	2023-04-16	mir pay	approved	2023-03-29
3	4	3	3002	3	2023-04-13	2023-04-14	mastercard	approved	2023-03-24
4	5	4	3000	16	2023-04-22	2023-04-23	visa	approved	2023-04-21
5	6	5	3001	5	2023-04-20	2023-04-26	visa	approved	2023-04-19
6	7	6	3002	24	2023-04-15	2023-04-24	visa	approved	2023-04-12
7	8	7	3002	2	2023-04-15	2023-04-26	visa	approved	2023-04-12
8	10	8	3002	21	2023-04-24	2023-05-10	mastercard	approved	2023-04-23
9	11	1	3000	13	2023-02-12	2023-02-26	visa	cancelled	2023-01-12
10	13	4	3002	4	2022-12-12	2022-12-16	visa	cancelled	2022-11-12

	id_order [PK] integer	client integer	employee integer	room integer	arrival_date date	departure_date date	payment_type text	status_order text	booking_date date
1	1	1	3000	10	2023-04-14	2023-04-16	visa	approved	2023-03-20
2	3	2	3001	6	2023-04-14	2023-04-16	mir pay	approved	2023-03-29
3	4	3	3002	3	2023-04-13	2023-04-14	mastercard	approved	2023-03-24
4	5	4	3000	16	2023-04-22	2023-04-23	visa	approved	2023-04-21
5	6	5	3001	5	2023-04-20	2023-04-26	visa	approved	2023-04-19
6	7	6	3002	24	2023-04-15	2023-04-24	visa	approved	2023-04-12
7	8	7	3002	2	2023-04-15	2023-04-26	visa	approved	2023-04-12
8	10	8	3002	21	2023-04-24	2023-05-10	mastercard	approved	2023-04-23
9	11	1	3000	13	2023-02-12	2023-02-26	visa	cancelled	2023-01-12

Создание индексов:

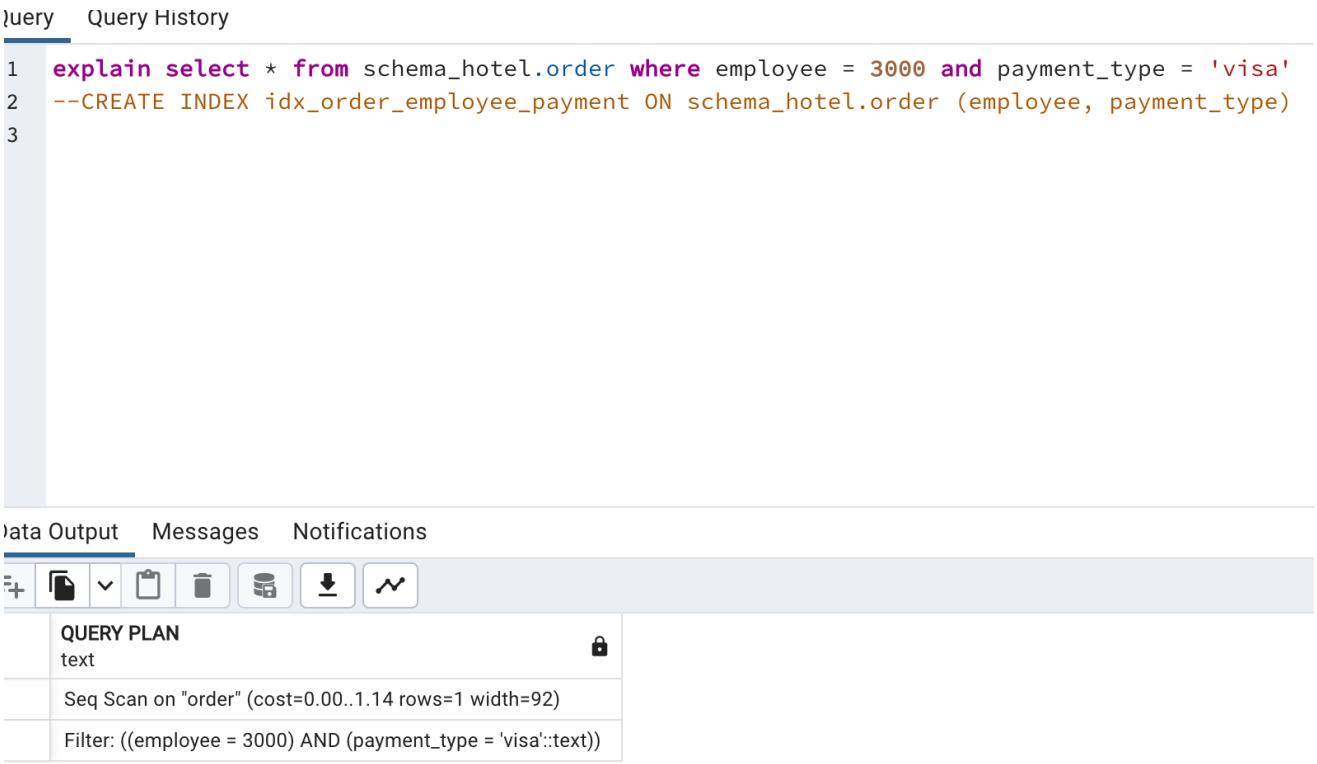
Простой индекс:

```
1  explain select * from schema_hotel.room where hotel = 3
2  --CREATE INDEX idx_rooms_hotel ON schema_hotel.room (hotel)
3
```

Data Output | Messages | Notifications

	QUERY PLAN text
1	Seq Scan on room (cost=0.00..1.30 rows=1 width=16)
2	Filter: (hotel = 3)

Составной индекс:



Выводы:

В данной лабораторной работе при выполнении варианта 1 я овладела практическими навыками создания представлений и запросов на выборку данных к базе данных PostgreSQL, использования подзапросов при модификации данных и индексов.