

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 № 4**  
по теме: «Запросы на выборку данных к БД PostgreSQL.  
ПРЕДСТАВЛЕНИЯ В PostgreSQL»  
по дисциплине: Проектирование и реализация баз данных

Специальность:  
45.03.04 Интеллектуальные системы в гуманитарной сфере

Проверил:  
Говорова М.М. \_\_\_\_\_  
Дата: «\_\_» \_\_\_\_\_ 20\_\_ г.  
Оценка \_\_\_\_\_

Выполнил:  
студентка группы  
К3242  
Абушаева М.И.

Санкт-Петербург 2021/2022

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

### 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 hotels

Схема логической модели базы данных, сгенерированная в Generate ERD



## 1. Запросы к базе данных

```
1  -- Составить список всех 2-местных номеров отелей,
2  -- с ценой менее 200 т.р., упорядочив данные в порядке уменьшения стоимости.
3  SELECT * FROM hotels.price JOIN hotels.room_types ON hotels.room_types.type_id = hotels.price.type_id
4  WHERE hotels.room_types.description LIKE '%двухместный%' AND hotels.price.price <= 200000 ORDER BY price desc
5
6  -- Выбрать все записи регистрации постояльцев, которые выехали из отелей в течение двух последних недель.
7  EXPLAIN SELECT * FROM hotels.clients JOIN hotels.residence ON hotels.residence.client_id = hotels.clients.client_id
8  WHERE hotels.residence.client_id IN (SELECT hotels.residence.client_id FROM hotels.residence
9  WHERE hotels.residence.checkout_date BETWEEN current_date - '14 days'::interval
10  AND current_date)
11
12  -- Определить, в каком отеле имеется наибольшее количество незанятых номеров на текущие сутки.
13  select distinct hotel_id, COUNT (distinct hotels.rooms.room_number) AS c
14  from hotels.rooms, hotels.residence
15  where hotels.rooms.room_number NOT IN
16  (select distinct hotels.residence.room_number from hotels.rooms, hotels.residence where current_date BETWEEN hotels.residence.checkin_date
17  AND hotels.residence.checkout_date AND hotels.rooms.room_number = hotels.residence.room_number)
18  GROUP BY hotel_id
19  HAVING COUNT (distinct hotels.rooms.room_number) = (SELECT MAX (c) FROM (select distinct hotel_id, COUNT (distinct hotels.rooms.room_number)
20  AS c
21  from hotels.rooms, hotels.residence where hotels.rooms.room_number NOT IN
22  (select distinct hotels.residence.room_number from hotels.rooms,
23  hotels.residence where current_date BETWEEN hotels.residence.checkin_date
24  AND hotels.residence.checkout_date
25  AND hotels.rooms.room_number = hotels.residence.room_number)
26  GROUP BY hotel_id) AS c)
27
28
29  -- Определить самый популярный тип номеров за последний год.
30  select hotels.rooms.type_id, COUNT (hotels.rooms.type_id) as popular from hotels.residence, hotels.rooms
31  where checkin_date BETWEEN 'now'::timestamp - '1 year'::interval AND current_date AND checkout_date
32  BETWEEN 'now'::timestamp - '1 year'::interval AND current_date AND hotels.residence.room_number = hotels.rooms.room_number GROUP BY hotels.rooms.type_id
33  HAVING COUNT(hotels.rooms.type_id) = (SELECT MAX (popular) FROM (select hotels.rooms.type_id, COUNT (hotels.rooms.type_id) as popular
34  from hotels.residence, hotels.rooms where checkin_date
35  BETWEEN 'now'::timestamp - '1 year'::interval AND current_date
36  and checkout_date BETWEEN 'now'::timestamp - '1 year'::interval AND current_date
37  and hotels.residence.room_number = hotels.rooms.room_number
38  GROUP BY hotels.rooms.type_id)
39  as popular)
40
41  -- Чему равен общий доход каждого отеля за последний месяц?
42
43  SELECT hotels.rooms.hotel_id,
44
45  CASE
46  WHEN (checkout_date < CURRENT_DATE)
47  THEN SUM((checkout_date - checkin_date) * hotels.price.price)
48  ELSE SUM((checkout_date - (checkout_date - CURRENT_DATE) - checkin_date) * hotels.price.price)
49  END
50
51  from hotels.residence, hotels.price, hotels.rooms
52  where hotels.residence.checkin_date BETWEEN hotels.price.start_date AND hotels.price.end_date
53  and hotels.residence.checkin_date BETWEEN 'now'::timestamp - '1 month'::interval AND CURRENT_DATE
54  and hotels.residence.room_number IN (hotels.rooms.room_number)
55  and hotels.rooms.type_id IN (hotels.price.type_id) GROUP BY hotels.rooms.hotel_id, checkout_date
56
57
58  -- Составить список свободных номеров на текущий день.
59  select distinct hotels.rooms.room_number, hotel_id from hotels.rooms, hotels.residence
60  where hotels.rooms.room_number NOT IN
61  (select distinct hotels.residence.room_number from hotels.rooms, hotels.residence
62  where current_date BETWEEN hotels.residence.checkin_date AND hotels.residence.checkout_date
63  and hotels.rooms.room_number = hotels.residence.room_number) AND hotel_id = 218978
64
65  -- Найти общие потери от незанятых номеров за текущий день по всей сети.
66  SELECT SUM(daily_loss) as result FROM (SELECT distinct hotels.rooms.room_number, SUM (distinct hotels.price.price)
67  as daily_loss from hotels.rooms, hotels.residence, hotels.price
68  where hotels.rooms.type_id = hotels.price.type_id
69  and current_date BETWEEN hotels.price.start_date AND hotels.price.end_date
70  and hotels.rooms.room_number NOT IN (select distinct
71  hotels.residence.room_number
72  from hotels.rooms, hotels.residence
73  where current_date BETWEEN hotels.residence.checkin_date
74  AND hotels.residence.checkout_date
75  AND hotels.rooms.room_number = hotels.residence.room_number)
76  group by hotels.rooms.room_number) as daily_loss
```

## 2. Представления

```
1  -- Для владельца компании (информация о доходах каждого отеля в сети за прошедший месяц).
2
3  CREATE VIEW hotels.month_earnings AS
4  select hotels.rooms.hotel_id,
5  CASE
6      WHEN (checkout_date < CURRENT_DATE)
7      THEN SUM((checkout_date - checkin_date) * hotels.price.price)
8      ELSE SUM((checkout_date - (checkout_date - CURRENT_DATE) - checkin_date) * hotels.price.price)
9  END
10
11 from hotels.residence, hotels.price, hotels.rooms
12 where hotels.residence.checkin_date BETWEEN hotels.price.start_date AND hotels.price.end_date
13 and hotels.residence.checkin_date BETWEEN 'now'::timestamp - '1 month'::interval AND CURRENT_DATE
14 and hotels.residence.room_number IN (hotels.rooms.room_number)
15 and hotels.rooms.type_id IN (hotels.price.type_id) GROUP BY hotels.rooms.hotel_id, checkout_date
16
17 SELECT * FROM hotels.vacant
18
19 -- Для турагентов (поиск свободных номеров в отелях).
20
21 CREATE VIEW hotels.vacant AS
22
23 select distinct hotels.rooms.room_number, hotel_id from hotels.rooms, hotels.residence
24 where hotels.rooms.room_number NOT IN
25 (select distinct hotels.residence.room_number from hotels.rooms, hotels.residence
26 where current_date BETWEEN hotels.residence.checkin_date AND hotels.residence.checkout_date
27 and hotels.rooms.room_number = hotels.residence.room_number)
28
```

### 3. Запросы на модификацию данных

#### 1) Добавить договор для нового сотрудника

```
1 -- добавить договор для нового сотрудника
2
3 INSERT INTO hotels.employment_contracts(
4     worker_id, hotel_id, contract_id, starting_date, contract_length_months, salary, "position")
5     SELECT (SELECT worker_id FROM hotels.workers WHERE surname = 'Михеев'), 218978, 1111, current_date, 6, 30000, 'Уборщик';
6
```

ДО:

	worker_id numeric	hotel_id numeric	contract_id numeric	starting_date date	contract_length_months numeric	salary numeric	position character varying
1	1	257350	2546	2020-04-24	13	50000	СММ-менеджер
2	2	257350	3547	2021-01-20	22	77000	Шеф-повар
3	3	257350	4568	2021-08-06	6	68000	Шофер
4	3	257350	4569	2022-02-06	18	45000	Шофер
5	4	218978	7158	2021-09-18	18	48000	Администратор
6	5	218978	2403	2021-12-25	15	43000	Уборщик
7	6	218978	8930	2021-10-05	24	21000	Аниматор
8	7	896167	23	2021-01-05	8	60000	Администратор
9	7	896167	9227	2021-08-05	3	75000	Старший Администратор
10	7	896167	7966	2021-11-05	11	35000	Администратор
11	8	896167	5216	2021-01-01	8	12000	Работник химчистки
12	8	896167	1224	2021-08-01	16	19000	Работник химчистки
13	9	896167	1781	2020-11-02	19	40000	Оператор колл-центра
14	100	257350	6776	2022-03-23	23	28000	Администратор

ПОСЛЕ:

1	1	257350	2546	2020-04-24	13	50000	СММ-менеджер
2	2	257350	3547	2021-01-20	22	77000	Шеф-повар
3	3	257350	4568	2021-08-06	6	68000	Шофер
4	3	257350	4569	2022-02-06	18	45000	Шофер
5	4	218978	7158	2021-09-18	18	48000	Администратор
6	5	218978	2403	2021-12-25	15	43000	Уборщик
7	6	218978	8930	2021-10-05	24	21000	Аниматор
8	7	896167	23	2021-01-05	8	60000	Администратор
9	7	896167	9227	2021-08-05	3	75000	Старший Администратор
10	7	896167	7966	2021-11-05	11	35000	Администратор
11	8	896167	5216	2021-01-01	8	12000	Работник химчистки
12	8	896167	1224	2021-08-01	16	19000	Работник химчистки
13	9	896167	1781	2020-11-02	19	40000	Оператор колл-центра
14	100	257350	6776	2022-03-23	23	28000	Администратор
15	10	218978	1111	2022-06-26	6	30000	Уборщик

#### 2) Повысить зарплату работнику который зарегистрировал больше всего гостей

```
-- повысить на 5000 т.р. зарплату работнику, который зарегистрировал больше всего гостей
UPDATE hotels.employment_contracts
SET salary = salary + 5000
WHERE worker_id IN (SELECT worker_id
                    FROM hotels.residence GROUP BY worker_id ORDER BY COUNT(res_id) DESC LIMIT 1);
```

ДО в пункте выше

	worker_id numeric	hotel_id numeric	contract_id numeric	starting_date date	contract_length_months numeric	salary numeric	position character varying
1	1	257350	2546	2020-04-24	13	50000	СММ-менеджер
2	2	257350	3547	2021-01-20	22	77000	Шеф-повар
3	3	257350	4568	2021-08-06	6	68000	Шофер
4	3	257350	4569	2022-02-06	18	45000	Шофер
5	5	218978	2403	2021-12-25	15	43000	Уборщик
6	6	218978	8930	2021-10-05	24	21000	Аниматор
7	7	896167	23	2021-01-05	8	60000	Администратор
8	7	896167	9227	2021-08-05	3	75000	Старший Администратор
9	7	896167	7966	2021-11-05	11	35000	Администратор
10	8	896167	5216	2021-01-01	8	12000	Работник химчистки
11	8	896167	1224	2021-08-01	16	19000	Работник химчистки
12	9	896167	1781	2020-11-02	19	40000	Оператор колл-центра
13	100	257350	6776	2022-03-23	23	28000	Администратор
14	10	218978	1111	2022-06-26	6	30000	Уборщик
15	4	218978	7158	2021-09-18	18	53000	Администратор









ПОСЛЕ:

### 3) Удалить неактивные договоры найма

```
-- удалить неактивные договоры найма
DELETE FROM hotels.employment_contracts
WHERE contract_id NOT IN (SELECT contract_id FROM hotels.employment_contracts WHERE current_date
BETWEEN hotels.employment_contracts.starting_date
AND (hotels.employment_contracts.starting_date
+ (hotels.employment_contracts.contract_length_months || ' months')
::INTERVAL)::DATE);
```

До выше

ПОСЛЕ

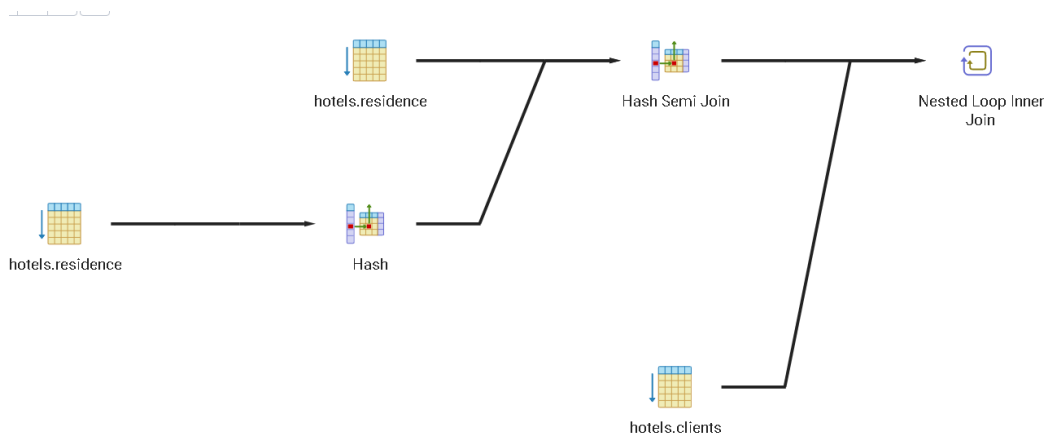
Data Output		Explain	Messages	Notifications			
	worker_id numeric 	hotel_id numeric 	contract_id numeric 	starting_date date 	contract_length_months numeric 	salary numeric 	position character varying 
1	2	257350	3547	2021-01-20	22	77000	Шеф-повар
2	3	257350	4569	2022-02-06	18	45000	Шофер
3	5	218978	2403	2021-12-25	15	43000	Уборщик
4	6	218978	8930	2021-10-05	24	21000	Аниматор
5	7	896167	7966	2021-11-05	11	35000	Администратор
6	8	896167	1224	2021-08-01	16	19000	Работник химчистки
7	100	257350	6776	2022-03-23	23	28000	Администратор
8	10	218978	1111	2022-06-26	6	30000	Уборщик
9	4	218978	7158	2021-09-18	18	53000	Администратор

## 4. Создание индексов

1) CREATE INDEX rti ON hotels.room\_types(type\_id)

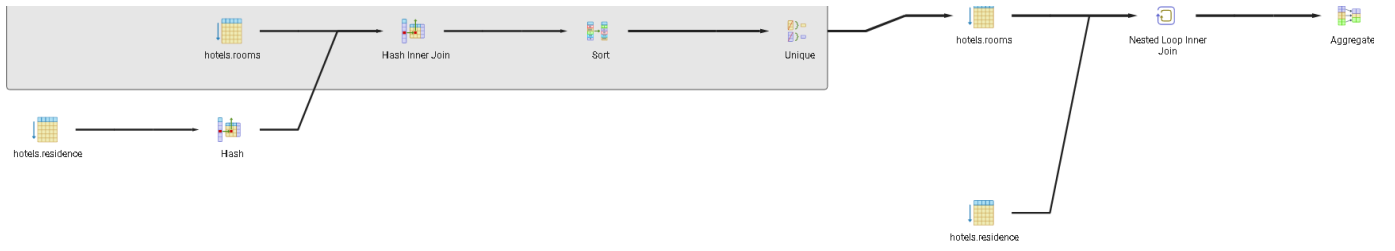
Время выполнения запроса 2 без индекса 59 msec

С индексом 40 msec



2) CREATE UNIQUE INDEX rsix ON hotels.residence(checkin\_date, room\_number)

Запрос 6. С индексом 67 msec. Без – 46 msec.



**Вывод:** при выполнении работы были составлены запросы к БД hotels, в том числе были выполнены запросы на создание представлений, а также запросы на модификацию данных с подзапросами, исследованы индексы.