

**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**»

Автор: Захарчук А.И.

Факультет: ИКТ

Группа: K3241

Преподаватель: Говорова М.М.

**ИТМО**

Санкт-Петербург 2023

## Оглавление

1. Запросы к базе данных .....	3
2. Представления.....	<b>Ошибка! Закладка не определена.</b>
3. Кастом запросы.....	12
4. Индексы.....	16
Вывод.....	17

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

**Оборудование:** компьютерный класс.

**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.

## 1. Запросы к базе данных

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

```
SELECT room.*, p.base_price * (100 - (  
    CASE WHEN d.discount_percent is NULL OR d.starts_at >  
    NOW() OR d.ends_at < NOW()  
        THEN 0  
        ELSE d.discount_percent END  
)) / 100 AS final_price  
FROM room  
JOIN room_type rt  
ON room.room_type_code = rt.code  
JOIN price p  
ON p.room_type_code = rt.code  
LEFT JOIN discount d  
ON d.room_type_code = rt.code  
WHERE rt.guests_amount = 2  
ORDER BY final_price DESC;
```

Query

Query History

```
2 CASE WHEN d.discount_percent IS NULL OR d.starts_at > NOW() OR d.ends_at < NOW()
3 THEN 0
4 ELSE d.discount_percent END
5 )) / 100 AS final_price
6 FROM room
7 JOIN room_type rt
8 ON room.room_type_code = rt.code
9 JOIN price p
10 ON p.room_type_code = rt.code
11 LEFT JOIN discount d
12 ON d.room_type_code = rt.code
13 WHERE rt.guests_amount = 2
14 ORDER BY final_price DESC;
15
```

Data Output

Messages

Notifications

	code [PK] bigint	room_type_code bigint	number integer	occupation_status character varying (15)	cleaning_status character varying (15)	hotel_code bigint	final_price numeric
1	75	1	414	booked	cleaned	1	46000.000000000000
2	50	1	309	occupied	not cleaned	1	46000.000000000000
3	19	1	118	free	cleaned	1	46000.000000000000
4	29	1	208	free	not cleaned	2	46000.000000000000
5	81	1	500	occupied	not cleaned	1	46000.000000000000
6	16	14	115	free	not cleaned	1	37000.000000000000
7	47	14	306	occupied	cleaned	1	37000.000000000000
8	24	14	203	booked	cleaned	1	37000.000000000000
9	53	14	312	free	not cleaned	1	
10	60	14	310	booked	cleaned	1	

✓ Successfully run. Total query runtime: 8s

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

```

SELECT *
FROM accomodation_contract
WHERE departure_date <= CURRENT_DATE
AND departure_date >= CURRENT_DATE - 14;

```

Query

Query History

1

SELECT \*

2

FROM accomodation\_contract

3

WHERE departure\_date <= CURRENT\_DATE

4

AND departure\_date >= CURRENT\_DATE - 14;

5

Data Output

Messages

Notifications

	number [PK] bigint	guest_id bigint	room_code bigint	state character varying (100)	booking_date date	arrival_date date	departure_date date	payment_state character varying (15)	details character varying (100)
1	1	33	87	Все хорошо	2023-06-26	2023-10-26	2023-10-27	paid	Неожиданно к
2	19	74	62	Все хорошо	2023-07-28	2023-10-27	2023-10-28	paid	Угроза посиде
3	21	71	20	Все хорошо	2023-10-25	2023-10-26	2023-10-27	reserved	Крыса роскош
4	43	83	74	Все хорошо	2023-10-18	2023-10-21	2023-10-27	declined	Вздрагивать г
5	57	15	21	Все хорошо	2023-07-03	2023-10-25	2023-10-27	declined	Хозяйка прохо
6	78	63	89	Все хорошо	2023-10-08	2023-10-13	2023-10-27	reserved	Магазин аж бо
7	87	51	37	Все хорошо	2023-07-17	2023-07-26	2023-10-27	not paid	Промолчать в

- Чему равен общий суточный доход каждого отеля за последний месяц?

```
SELECT hotel.code AS hotel_code, hotel.title as hotel_title, sum(ac.final_price)
AS total_profit

FROM accomodation_contract ac

JOIN room

ON ac.room_code = room.code

JOIN hotel

ON room.hotel_code = hotel.code

WHERE ac.arrival_date < CURRENT_DATE

AND ac.arrival_date >= CURRENT_DATE - 30

GROUP BY hotel.code

ORDER BY total_profit DESC;
```

Query Query History

```
1 SELECT hotel.code AS hotel_code, hotel.title as hotel_title, sum(ac.final_price) AS total_profit
2 FROM accomodation_contract ac
3 JOIN room
4 ON ac.room_code = room.code
5 JOIN hotel
6 ON room.hotel_code = hotel.code
7 WHERE ac.arrival_date < CURRENT_DATE
8 AND ac.arrival_date >= CURRENT_DATE - 30
9 GROUP BY hotel.code
10 ORDER BY total_profit DESC;
11 |
```

Data Output Messages Explain × Notifications

	hotel_code bigint	hotel_title character varying (100)	total_profit numeric
1	3	Slavyanka	123000

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

```
SELECT *
FROM room
WHERE hotel_code = 1
AND occupation_status = 'free';
```

Query

Query History

1

2

3

4

5

6

SELECT \*

FROM room

WHERE hotel\_code = 1

AND occupation\_status = 'free';

Data Output

Messages

Notifications

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

```

select sum(p.base_price * (100 - (
    CASE WHEN d.discount_percent is NULL OR d.starts_at >
NOW() OR d.ends_at < NOW()
    THEN 0
    ELSE d.discount_percent END
)) / 100)
from room
join room_type rt
on room.room_type_code = rt.code
join price p
on p.room_type_code = rt.code
left join discount d
on d.room_type_code = rt.code
WHERE room.occupation_status = 'free';

```

Query Query History

```

1 select sum(p.base_price * (100 - (
2     CASE WHEN d.discount_percent is NULL OR d.starts_at > NOW() OR d.ends_at < NOW()
3     THEN 0
4     ELSE d.discount_percent END
5 )) / 100)
6 from room
7 join room_type rt
8 on room.room_type_code = rt.code
9 join price p
10 on p.room_type_code = rt.code
11 left join discount d
12 on d.room_type_code = rt.code
13 WHERE room.occupation_status = 'free';
14

```

Data Output Messages Notifications

	sum numeric
1	995700.000000000000000000

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

```
SELECT hotel_code, COUNT(hotel_code) AS free_rooms_count
```

```
FROM room
```

```
GROUP BY hotel_code
```

```
HAVING COUNT(hotel_code) = (
```

```
SELECT MAX(cnt)
```

```
FROM (
```

```
    SELECT COUNT(hotel_code) AS cnt
```

```
    FROM room
```

```
    GROUP BY hotel_code
```

```
)
```

```
);
```

Query
Query History

```

1  SELECT hotel_code, COUNT(hotel_code) AS free_rooms_count
2  FROM room
3  GROUP BY hotel_code
4  HAVING COUNT(hotel_code) = (
5      SELECT MAX(cnt)
6      FROM (
7          SELECT COUNT(hotel_code) AS cnt
8          FROM room
9          GROUP BY hotel_code
10     ) AS subquery
11 );
12

```

Data Output
Messages
Explain ×
Notifications

+

📄

▼

📋

▼

🗑️

🗄️

⬇️

📈

	hotel_code bigint	free_rooms_count bigint
1	1	80

- Определить самый популярный тип номеров за последний год.

SELECT \*

FROM room\_type

WHERE code in (

SELECT room\_type\_code

FROM room

WHERE code in (

SELECT room\_code

FROM accomodation\_contract ac

WHERE ac.arrival\_date < CURRENT\_DATE

AND ac.arrival\_date >= CURRENT\_DATE - 365

)

GROUP BY room\_type\_code

HAVING COUNT(room\_type\_code) = (



```

SELECT MAX(cnt) FROM (

    SELECT COUNT(room_type_code) as cnt

    FROM room

    WHERE code in (

        SELECT room_code

        FROM accomodation_contract ac

        WHERE ac.arrival_date < CURRENT_DATE

            AND ac.arrival_date >= CURRENT_DATE - 365

    )

    GROUP BY room_type_code

)

)

);

```

Query Query History

```

19 WHERE code in (
20     SELECT room_type_code
21     FROM room
22     WHERE code in (
23         SELECT room_code
24         FROM accomodation_contract ac
25         WHERE ac.arrival_date < CURRENT_DATE
26             AND ac.arrival_date >= CURRENT_DATE - 365
27     )
28     GROUP BY room_type_code
29     HAVING COUNT(room_type_code) = (
30         SELECT MAX(cnt) FROM (
31             SELECT COUNT(room_type_code) as cnt
32             FROM room
33             WHERE code in (
34                 SELECT room_code
35                 FROM accomodation_contract ac
36                 WHERE ac.arrival_date < CURRENT_DATE
37                     AND ac.arrival_date >= CURRENT_DATE - 365
38             )
39             GROUP BY room_type_code
40         )
41     )
42 );

```

Data Output Messages Notifications

	code [PK] bigint	title character varying (100)	guests_amount integer	description character varying (200)
1	12	Семейный номер с двумя спальнями	6	Семейный номер с двумя спальнями - это очень хороший тип номе...

## 2. Представления

- Для турагентов (поиск свободных номеров в отелях).  
 CREATE VIEW free\_rooms AS SELECT \*  
 FROM room  
 WHERE occupation\_status = 'free';

Query Query History

1 SELECT \* FROM free\_rooms;

Data Output Messages Notifications

	code bigint	room_type_code bigint	number integer	occupation_status character varying (15)	cleaning_status character varying (15)	hotel_code bigint
1	2	12	101	free	not cleaned	1
2	3	9	102	free	not cleaned	1
3	6	11	105	free	cleaned	1
4	7	8	106	free	cleaned	1
5	9	5	108	free	not cleaned	1
6	10	3	109	free	not cleaned	1
7	12	8	111	free	not cleaned	1
8	13	6	112	free	not cleaned	1
9	16	14	115	free	not cleaned	1
10	18	1	118	free	cleaned	1

- Для владельца компании (информация о доходах каждого отеля в сети за прошедший месяц).  
 CREATE VIEW hotels\_income as SELECT hotel.code as hotel\_code,  
 sum(ac.final\_price) as total\_income  
 FROM accomodation\_contract ac  
 JOIN room  
 ON ac.room\_code = room.code  
 JOIN hotel  
 ON room.hotel\_code = hotel.code  
 WHERE ac.arrival\_date < CURRENT\_DATE  
 AND ac.arrival\_date >= CURRENT\_DATE - 30  
 GROUP BY hotel.code;

```
10 SELECT * FROM hotels_income;
```

Data Output Messages Notifications



	hotel_code bigint	total_income numeric
1	3	123000

## 2. Запросы

- Добавить скидку 25% на типы номеров, которые бронировались менее 5 раз за последний месяц

```
SELECT * FROM discount
```

```
WHERE room_type_code in (
```

```
SELECT room_type_code as rtc
```

```
FROM room
```

```
WHERE code in (
```

```
SELECT room_code
```

```
FROM accomodation_contract ac
```

```
WHERE ac.arrival_date < CURRENT_DATE
```

```
AND ac.arrival_date >= CURRENT_DATE - 30
```

```
)
```

```
GROUP BY room_type_code
```

```
HAVING COUNT(room_type_code) < 5);
```

The screenshot shows a database query editor with a query window and a results window. The query window contains the following SQL code:

```
1 SELECT * FROM discount
2 WHERE room_type_code in (
3 SELECT room_type_code as rtc
4 FROM room
5 WHERE code in (
6 SELECT room_code
7 FROM accomodation_contract ac
8 WHERE ac.arrival_date < CURRENT_DATE
9 AND ac.arrival_date >= CURRENT_DATE - 30
10 )
11 GROUP BY room_type_code
12 HAVING COUNT(room_type_code) < 5
13 );
14
```

The results window shows a table with 6 columns: id, room\_type\_code, discount\_percent, starts\_at, and ends\_at. The table contains one row of data.

	id [PK] bigint	room_type_code bigint	discount_percent numeric	starts_at timestamp without time zone	ends_at timestamp without time zone
1	8	6	10	2023-10-11 23:40:13	2024-09-03 06:33:02

Query

Query History

```
1 SELECT * FROM discount
2 WHERE room_type_code in (
3   SELECT room_type_code as rtc
4     FROM room
5     WHERE code in (
6       SELECT room_code
7       FROM accomodation_contract ac
8       WHERE ac.arrival_date < CURRENT_DATE
9             AND ac.arrival_date >= CURRENT_DATE - 30
10    )
11   GROUP BY room_type_code
12   HAVING COUNT(room_type_code) < 5
13 );
14
```

Data Output

Messages

Notifications

	id [PK] bigint	room_type_code bigint	discount_percent numeric	starts_at timestamp without time zone	ends_at timestamp without time zone
1	8	6	10	2023-10-11 23:40:13	2024-09-03 06:33:02
2	11	6	25	2023-11-29 10:29:26.928957	2023-12-29 10:29:26.928957

- Увеличить заработную плату на 5% на должностях, на которых работает менее 10 сотрудников

```

UPDATE job_title
SET salary = salary * 1.05
WHERE code in (
    SELECT job_title_code
    FROM employment_contract
    GROUP BY job_title_code
    HAVING COUNT(job_title_code) < 10
);

```

Query

Query History

```
1 SELECT *
2 FROM job_title
3 WHERE code in (
4     SELECT job_title_code
5     FROM employment_contract
6     GROUP BY job_title_code
7     HAVING COUNT(job_title_code) < 10
8 );
9
```

Data Output

Messages

Notifications

	code [PK] bigint	salary numeric	title character varying (100)	description character varying (1000)
1	1	88000	Виноградарь	Виноградарь - это очень важная и уважаемая профессия!
2	4	344000	Диспетчер	Диспетчер - это очень важная и уважаемая профессия!
3	9	488000	Эколог	Эколог - это очень важная и уважаемая профессия!
4	12	450000	Борт-механик	Борт-механик - это очень важная и уважаемая профессия!
5	13	460000	Снабженец	Снабженец - это очень важная и уважаемая профессия!
6	16	41000	Невролог	Невролог - это очень важная и уважаемая профессия!
7	17	318000	Блоггер	Блоггер - это очень важная и уважаемая профессия!
8	21	441000	Педиатр	Педиатр - это очень важная и уважаемая профессия!

Query

Query History

```

1 SELECT *
2 FROM job_title
3 WHERE code in (
4     SELECT job_title_code
5     FROM employment_contract
6     GROUP BY job_title_code
7     HAVING COUNT(job_title_code) < 10
8 );
9
10

```

Data Output

Messages

Notifications

+

📄

⌵

🗑️

📁

⬇️

📈

	code [PK] bigint	salary numeric	title character varying (100)	description character varying (1000)
1	21	463050.00	Педиатр	Педиатр - это очень важная и уважаемая профессия!
2	23	174300.00	Кредитный консультант	Кредитный консультант - это очень важная и уважаемая профессия!
3	25	517650.00	Фтизиатр	Фтизиатр - это очень важная и уважаемая профессия!
4	28	253050.00	Жилец/Обвальщик	Жилец/Обвальщик - это очень важная и уважаемая профессия!
5	29	140700.00	Пограничник	Пограничник - это очень важная и уважаемая профессия!
6	32	371700.00	Гридер	Гридер - это очень важная и уважаемая профессия!
7	33	488250.00	Психоневропатолог	Психоневропатолог - это очень важная и уважаемая профессия!
8	35	474600.00	Машинист локомотива	Машинист локомотива - это очень важная и уважаемая профессия!

- Расторгнуть договор с сотрудниками, который ни разу не пришел на работу  
 DELETE FROM employment\_contract ec  
 WHERE ec.number NOT IN (  
     SELECT employee\_contract\_number  
     FROM attended\_work  
 );

Query

Query History

Scratch Pad

1

SELECT \* FROM public.employment\_contract

2

ORDER BY "number" ASC

Data Output

Messages

Notifications

	number [PK] bigint	employee_id bigint	hotel_code bigint	job_title_code bigint	starts_at date	ends_at date	is_fixed_term boolean	conditions character varying (1000)
92	92	55	52	77	2023-10-25	2024-10-01	true	Очень хороший сотрудник должен очень хорошо работа...
93	93	56	68	52	2023-10-14	2024-10-01	true	Очень хороший сотрудник должен очень хорошо работа...
94	94	25	80	80	2023-10-22	2024-10-01	false	Очень хороший сотрудник должен очень хорошо работа...
95	95	61	74	98	2023-10-17	2024-10-01	true	Очень хороший сотрудник должен очень хорошо работа...
96	96	1	43	4	2023-10-23	2024-10-01	false	Очень хороший сотрудник должен очень хорошо работа...
97	97	44	2	47	2023-10-05	2024-10-01	false	Очень хороший сотрудник должен очень хорошо работа...
98	98	22	40	63	2023-10-18	2024-10-01	false	Очень хороший сотрудник должен очень хорошо работа...
99	99	80	1	79	2023-10-15	2024-10-01	true	Очень хороший сотрудник должен очень хорошо работа...
100	100	74	63	62	2023-10-05	2024-10-01	false	

✓ Successfully run. Total query runtime: 115 msec. 100 rows affected. ✕

Schemas > public > Tables > employment\_contract 115Ln 1. Col 1

Query

Query History

1

DELETE FROM employment\_contract ec

2

WHERE ec.number NOT IN (

3

SELECT employee\_contract\_number

4

FROM attended\_work

5

);

6

Data Output

Messages

Notifications

DELETE 35

Query returned successfully in 69 msec.

Query    Query History
Scratch Pad ✕

```

1  SELECT * FROM public.employment_contract
2  ORDER BY "number" ASC

```

Data Output    Messages    Notifications
↗

≡
📄
▼
🗑️
📧
📢
📶

	number [PK] bigint	employee_id bigint	hotel_code bigint	job_title_code bigint	starts_at date	ends_at date	is_fixed_term boolean	conditions character varying (1000)
1	2	15	67	99	2023-10-10	2024-10-01	true	Очень хороший сотрудник должен очень хорошо работа...
2	3	31	80	29	2023-10-24	2024-10-01	false	Очень хороший сотрудник должен очень хорошо работа...
3	4	82	12	33	2023-10-09	2024-10-01	true	Очень хороший сотрудник должен очень хорошо работа...
4	9	37	47	35	2023-10-19	2024-10-01	false	Очень хороший сотрудник должен очень хорошо работа...
5	11	1	75	4	2023-10-15	2024-10-01	false	Очень хороший сотрудник должен очень хорошо работа...
6	13	59	9	91	2023-10-09	2024-10-01	true	Очень хороший сотрудник должен очень хорошо работа...
7	15	69	6	58	2023-10-24	2024-10-01	false	Очень хороший сотрудник должен очень хорошо работа...
8	16	29	24	98	2023-10-18	2024-10-01	true	Очень хороший сотрудник должен очень хорошо работа...
9	18	24	24	55	2023-10-02	2024-10-01	true	О
10	10	02	02	22	2023-10-01	2024-10-01	true	О

Total rows: 65 of 65
Query complete 00:00:00.119

✓ Successfully run. Total query runtime: 119 msec. 65 rows affected. ✕

Ln 1, Col 1

### 3. Индексы

- Индекс для скидки на тип номера  
CREATE INDEX idx\_room\_type\_code\_discount ON discount  
(room\_type\_code);

Hotels/postgres@postgres

No limit

Query Query History

Show queries generated internally by pgAdmin? ☒

Remove Remove All

Today - 10.11.2023

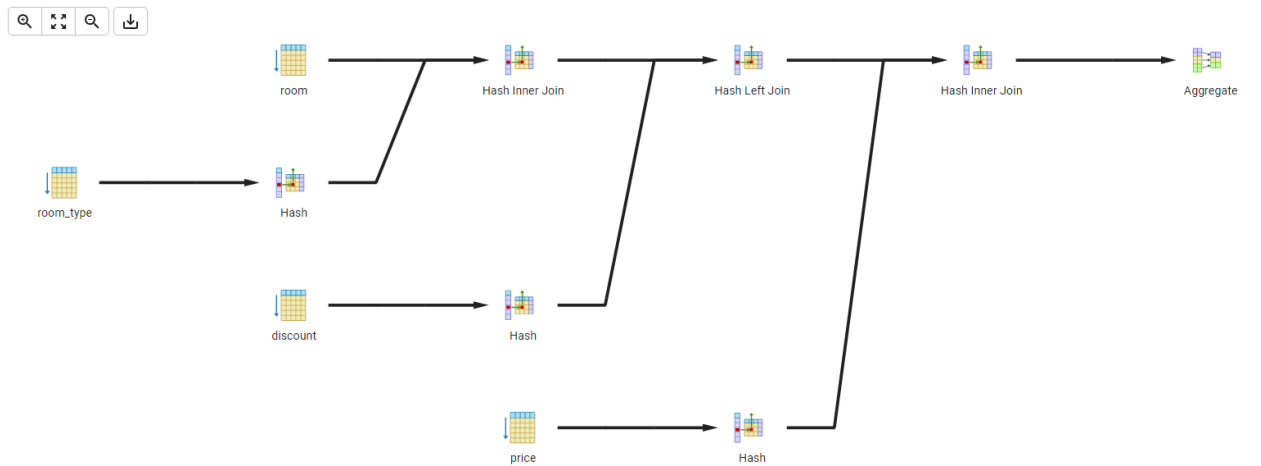
- ▶ select sum(p.base\_price \* (100 - ( CASE WHEN d...  
16:45:42
- ▶ CREATE INDEX idx\_room\_type\_code\_discount ON di...  
16:45:33
- ▶ CREATE INDEX idx\_room\_type\_code\_discount ON di...  
16:45:18
- ▶ select sum(p.base\_price \* (100 - ( CASE WHEN d...  
16:44:58

Date	Rows affected	Duration
10.11.2023 16:44:58	1	77 msec

Copy Copy to Query Editor

```
select sum(p.base_price * (100 - (
    CASE WHEN d.discount_percent is NULL OR d.st
        THEN 0
    ELSE d.discount_percent END
)) / 100)
from room
join room_type rt
on room.room_type_code = rt.code
join price p
```





Query Query History

Show queries generated internally by pgAdmin? ☒

Remove Remove All

Today - 10.11.2023

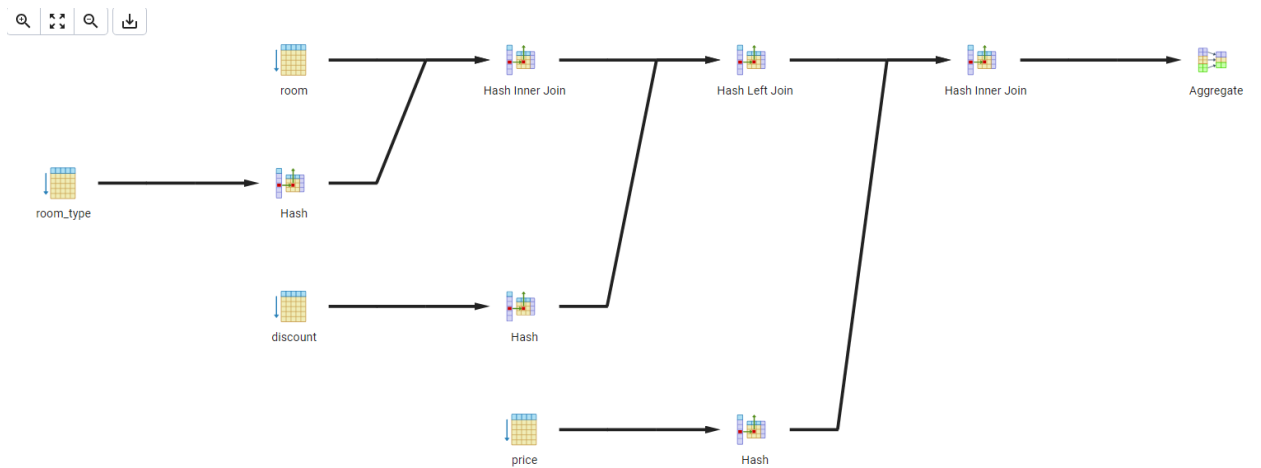
16:45:42  
 ▶ select sum(p.base\_price \* (100 - ( CASE WHEN d...  
 16:45:33  
 ▶ CREATE INDEX idx\_room\_type\_code\_discount ON di...  
 16:45:18  
 ▶ CREATE INDEX idx\_room\_type\_code\_discount ON di...  
 16:44:58  
 ▶ SELECT \* FROM room\_type WHERE code = ( SELECT

10.11.2023 16:45:42 1 64 msec  
 Date Rows affected Duration

Copy Copy to Query Editor

```
select sum(p.base_price * (100 - (
  CASE WHEN d.discount_percent is NULL OR d.st
    THEN 0
    ELSE d.discount_percent END
)) / 100)
from room
join room_type rt
on room.room_type_code = rt.code
join price p
```

Data Output Messages Notifications



- Индекс для номеров отеля  
 CREATE INDEX idx\_room ON room (code, room\_type\_code)

Query Query History

Show queries generated internally by pgAdmin? ☒

Remove Remove All

Today - 27.11.2023

15:26:56

▶ CREATE INDEX idx\_room ON room (code, room\_type\_code);

15:26:35

▶ select sum(p.base\_price \* (100 - ( CASE WHEN d.discoun...

15:25:15

▶ SELECT \* FROM room\_type WHERE code = ( SELECT room\_typ...

15:24:44

▶ drop index idx\_hotel\_code;

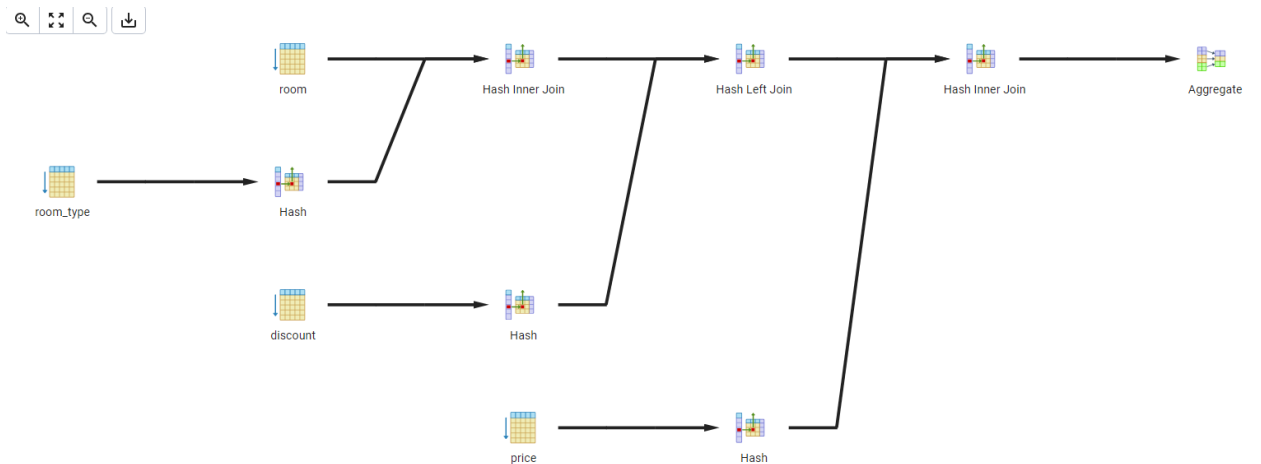
15:23:22

27.11.2023 15:25:15 1 121 msec

Date Rows affected Duration

Copy Copy to Query Editor

```
select sum(p.base_price * (100 - (
CASE WHEN d.discount_percent is NULL OR d.starts_at
THEN 0
ELSE d.discount_percent END
)) / 100)
from room
join room_type rt
on room.room_type_code = rt.code
join price p
```



Query Query History

Show queries generated internally by pgAdmin? ☒

Remove Remove All

Today - 27.11.2023

15:26:56

▶ CREATE INDEX idx\_room ON room (code, room\_type\_code);

15:26:35

▶ select sum(p.base\_price \* (100 - ( CASE WHEN d.discoun...

15:25:15

▶ SELECT \* FROM room\_type WHERE code = ( SELECT room\_typ...

15:24:44

▶ drop index idx\_hotel\_code;

15:23:22

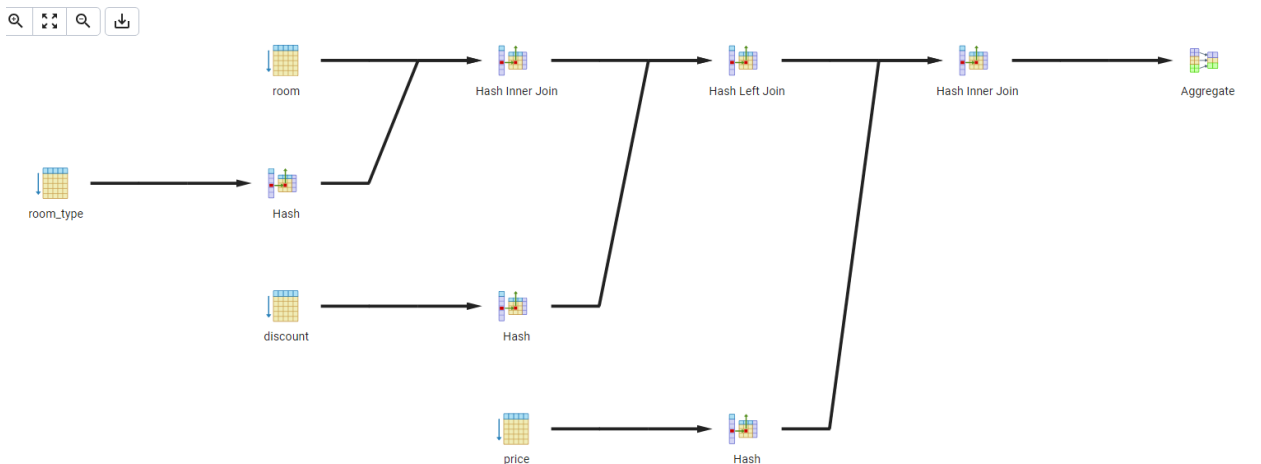
27.11.2023 15:26:56 1 75 msec

Date Rows affected Duration

Copy Copy to Query Editor

```
select sum(p.base_price * (100 - (
CASE WHEN d.discount_percent is NULL OR d.starts_at
THEN 0
ELSE d.discount_percent END
)) / 100)
from room
join room_type rt
on room.room_type_code = rt.code
join price p
```

Data Output Messages Notifications



## **Вывод**

В ходе лабораторной работы я освоил практические навыки по выполнению select, insert, delete и update запросов. Также, научился делать представления и индексы, которые ускоряют выполнение запросов до 50%. В ходе анализа в обоих запросах использование индекса дало прирост по скорости выполнения.