

**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.....	11
4. Индексы.....	15
Вывод.....	16

**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, 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, 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
Notifications

	code [PK] bigint	total_profit numeric
1	1	1423000
2	3	529000
3	2	428000

- 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 *
FROM hotel
ORDER BY (
    SELECT COUNT(code)
    FROM room
    WHERE hotel_code = hotel.code
    AND room.occupation_status = 'free'
) DESC
LIMIT 1;

```

Query
Query History

```

1  SELECT *
2  FROM hotel
3  ORDER BY (
4      SELECT COUNT(code)
5      FROM room
6      WHERE hotel_code = hotel.code
7      AND room.occupation_status = 'free'
8  ) DESC
9  LIMIT 1;
10

```

Data Output
Messages
Notifications

	code [PK] bigint	title character varying (100)	address character varying (100)
1	1	Nevskiy plaza	Санкт-Петербург, пр-т Невский, д. 1

- Определить самый популярный тип номеров за последний год.  

```

SELECT *
FROM room_type
WHERE code = (
    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
    ORDER BY COUNT(code) DESC
    LIMIT 1
);

```



Query Query History

```

3 WHERE code = (
4     SELECT room_type_code
5     FROM room
6     WHERE code in (
7         SELECT room_code
8         FROM accomodation_contract ac
9         WHERE ac.arrival_date < CURRENT_DATE
10            AND ac.arrival_date >= CURRENT_DATE - 365
11     )
12     GROUP BY room_type_code
13     ORDER BY COUNT(code) DESC
14     LIMIT 1
15 );
16

```

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	10	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, sum(ac.final_price)
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;
```

The screenshot shows a database query editor with two tabs: "Query" and "Query History". The "Query" tab is active, displaying a SQL query. Below the query editor, there are three tabs: "Data Output", "Messages", and "Notifications". The "Data Output" tab is active, showing a table with the results of the query. The table has two columns: "code" (bigint) and "sum" (numeric). The results are as follows:

	code bigint	sum numeric
1	3	529000
2	1	1423000
3	2	428000

## 2. Запросы

- Добавить новые отели  
`INSERT INTO hotel(title, address)`

`SELECT last_name, address`

`FROM guest`

`LIMIT 10;`

Query		Query History	
1	<code>SELECT * FROM public.hotel</code>		
2	<code>ORDER BY code ASC</code>		
Data Output			
Messages			
Notifications			
	code [PK] bigint	title character varying (100)	address character varying (100)
95	94	Frye	Санкт-Петербург, пр. Одесский, д. 7/4 к. 921
96	95	Waters	Санкт-Петербург, пер. Войкова, д. 231
97	96	Jones	Санкт-Петербург, алл. Минская, д. 94 стр. 725
98	97	Palmer	Санкт-Петербург, ш. Осенное, д. 91
99	98	Barker	Санкт-Петербург, алл. Дорожная, д. 1/1 стр. 7/6
100	99	Phillips	Санкт-Петербург, ш. Авиационное, д. 921 стр. 4
101	100	Jones	Санкт-Петербург, ш. Сплавное, д. 4 стр. 972
102	110	Венедикт	г. Туапсе, бул. Есенина, д. 5/1 стр. 31, 467317
103	111	Венедикт	г. Туапсе, бул. Есенина, д. 5/1 стр. 31, 467317

Query		Query History	
1	SELECT *	FROM public.hotel	
2	ORDER BY	code ASC	

  

Data Output		Messages		Notifications	
	code	title	address		
	[PK] bigint	character varying (100)	character varying (100)		
105	113	Радован	г. Шелагонцы, алл. Декабристов, д. 1, 471459		
106	114	Юрьевна	клх Орск, ул. Аэродромная, д. 8/4, 186058		
107	115	Владлен	п. Егорьевск, ш. Мичурина, д. 3, 906338		
108	116	Александрович	г. Арсеньев, ул. Февральская, д. 816 к. 599, 704708		
109	117	Силантий	к. Южноуральск, пер. Парковый, д. 9, 487951		
110	118	Анастасия	к. Токма, ш. Тимирязева, д. 80, 954521		
111	119	Евстафий	с. Ростов, наб. Чкалова, д. 4 стр. 66, 505784		
112	120	Зиновий	г. Каменск-Уральский, бул. Курский, д. 8 стр. 5, 4052...		
113	121	Бенедиктович	д. Камышин, ул. Пригородная, д. 803 к. 600, 372342		

- Поменять фамилии гостей на фамилии работников

UPDATE guest

SET last\_name = (

SELECT last\_name

FROM employee

WHERE guest.id = employee.id

);

QueryQuery History

1 SELECT \* FROM public.guest  
2 ORDER BY id ASC

Scratch Pad

Data OutputMessagesNotifications

	first_name character varying (50)	last_name character varying (50)	patronymic character varying (50)	address character varying (100)	date_of_birth date	id [PK] bigint	phone_number character varying (15)
92	Щукин	Марсович	Евсеевич	ст. Лодейное Поле, наб. Нагорная, д. 389 стр. 6/4, 002348	1969-09-11	92	+72215251742
93	Варвара	Устин	Русакова	г. Москва, МГУ, ш. Торговое, д. 622, 440742	1960-03-26	93	+7 424 210 9705
94	Никонов	Николаевна	Ааронович	с. Ярославль, бул. Ангарский, д. 891, 267903	1987-04-07	94	8 845 740 45 08
95	Егор	Ярослав	Андреев	г. Верецагино (Перм.), наб. Коммунальная, д. 759, 450004	1958-08-01	95	+7 (224) 294-28-4
96	Яков	Парфен	Николаев	д. Павловская, ул. Курганная, д. 6/6, 972917	1958-09-16	96	8 862 647 22 36
97	Август	Григорьевна	Моисеев	п. Дно, пр. Папанина, д. 13 к. 26, 381620	2005-06-24	97	8 (785) 656-92-64
98	Красильников	Анастасия	Тихонович	п. Нарьян-Мар, алл. Запорожская, д. 6 стр. 14, 523682	1968-11-14	98	+7 (822) 998-3510
99	Устинов	Эдуардович	Юлианович	д. Калевала, бул. 40 лет Октября, д. 93 стр. 94, 178448	1992-09-08	99	8 (543) 193-7511
100	Синклитикия	Артемона	Пономарева	п. Элиста, пер. Мира, д. 56 к. 9/2, 530822	1976-02-22	100	+78026702581

Total rows: 100 of 100Query complete 00:00:00.111

Data OutputMessagesNotifications

	first_name character varying (50)	last_name character varying (50)	patronymic character varying (50)	address character varying (100)	date_of_birth date	id [PK] bigint	phone_number character varying (15)
1	Данилов	Феликсович	Герасимович	г. Туапсе, бул. Есенина, д. 5/1 стр. 31, 467317	1962-10-21	1	8 (554) 319-55-44
2	Фролова	Олег	Леонидовна	к. Адигейск, пр. Снежный, д. 8 стр. 8/7, 603199	1975-11-23	2	+7 524 419 1335
3	Корнилов	Наталья	Герасимович	клх Костомукша, алл. Абрикосовая, д. 62 к. 8, 661209	1963-05-15	3	+7 (441) 597-12-4
4	Лавр	Олеговна	Федосеев	п. Глазов, ш. Детское, д. 279, 286670	1979-10-07	4	8 787 919 3254
5	Мясникова	Елисеевич	Святославовна	с. Руза, пер. Морозова, д. 4, 882512	1975-05-12	5	+7 195 838 52 62
6	Ангела	Оксана	Горшкова	г. Батайск, наб. Димитрова, д. 5, 749993	1969-09-14	6	8 900 644 71 80
7	Фортунат	Мироновна	Федоров	г. Шелагонцы, алл. Декабристов, д. 1			
8	Гришина	Артемьевич	Владимировна	клх Орск, ул. Аэродромная, д. 8/4, 18			
9	Галкин	Потап	Изотович	п. Егорьевск, ш. Мичурина, д. 3, 906338			

Successfully run. Total query runtime: 106 msec. 100 rows affected.

Query returned successfully in 64 msec.

- 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
)
LIMIT 1;

```

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	

Schemas > public > Tables > employment\_contract

115

Ln 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

```

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	0
10	10	03	03	22	2023-10-01	2024-10-01	true	0

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

Ln 1, Col 1

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

### 3. Индексы

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

Hotels/postgres@postgres

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

```

10.11.2023 16:44:58 1 77 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

```

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
- ▶ SELECT \* FROM room\_type WHERE code = ( SELECT

Data Output Messages Notifications

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
```

- Индекс для чего-то

Query Query History

Show queries generated internally by pgAdmin? ☒

Remove Remove All

Today - 10.11.2023

- ▶ SELECT hotel.code, sum(ac.final\_price) FROM accomodati...  
16:50:42
- ▶ CREATE INDEX idx\_code\_hotel ON hotel (code);  
16:50:33
- ▶ SELECT hotel.code, sum(ac.final\_price) FROM accomodati...  
16:49:40
- ▶ DROP INDEX idx\_room\_type\_code\_discount;  
16:49:27
- ▶ DROP INDEX idx\_room\_type\_code\_discount ON discount;

10.11.2023 16:49:40 3 70 msec  
Date Rows affected Duration

Copy Copy to Query Editor

```
SELECT hotel.code, sum(ac.final_price)
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;
```

Query Query History

Show queries generated internally by pgAdmin? ☒

Remove Remove All

Today - 10.11.2023

- ▶ SELECT hotel.code, sum(ac.final\_price) FROM accomodati...  
16:50:42
- ▶ CREATE INDEX idx\_code\_hotel ON hotel (code);  
16:50:33
- ▶ SELECT hotel.code, sum(ac.final\_price) FROM accomodati...  
16:49:40
- ▶ DROP INDEX idx\_room\_type\_code\_discount;  
16:49:27
- ▶ DROP INDEX idx\_room\_type\_code\_discount ON discount;

10.11.2023 16:50:42 3 66 msec  
Date Rows affected Duration

Copy Copy to Query Editor

```
SELECT hotel.code, sum(ac.final_price)
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;
```



## **Вывод**

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