

**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: Игнатьев А.А.

Факультет: ИКТ

Группа: K3240

Преподаватель: Говорова М.М.



Санкт-Петербург 2023

## Оглавление

Цель работы: .....	3
Практическое задание .....	3
Вариант 19. БД «Издательство компьютерной литературы».....	3
Выполнение.....	4

## **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 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.

## **Вариант 19. БД «Издательство компьютерной литературы»**

### **Описание предметной области:**

Описание предметной области: Издательство занимается выпуском литературы по различным областям ИТ. Покупатели (юридические лица) приобретают книги на базе издательства.

Когда на базе заканчиваются книги, издается дополнительный тираж.

В каждом заказе заказчик может заказать разную литературу. Для покупки заключается договор, который сопровождает менеджер издательства. По каждому проекту составляется договор с Заказчиком (в 2-х экземплярах для каждой стороны). По каждому договору оформляется два счета – на предоплату и остаток. После выполнения проекта подписывается Акт выполненных работ (в 2-х экземплярах для каждой стороны).

Каждое издание относится к определенной области ИТ, имеет тип (учебник, учебное пособие и т.п.), номер издания (если есть), может иметь одного или нескольких авторов, выпускаться под редакцией одного или нескольких авторов и т.п. При формировании списка авторов или списка “под редакцией” важен порядок авторов.

На каждое издание составляется Техническое задание, в котором могут участвовать несколько редакторов, один из которых является главным редактором. На каждую книгу может быть несколько ТЗ, в зависимости от переплета, типа бумаги, наличия иллюстраций и т.д.

БД должна содержать следующий минимальный набор сведений: Фамилия автора. Имя автора. Отчество автора. Код автора. E-mail автора. Код ISBN. Название книги. Количество страниц. Наличие иллюстраций. Код категории книги. Категория книги.

Количество страниц. Год начала издания. Розничная цена книги. Тираж. Дата тиража. Количество экземпляров на базе издательства. Код заказчика. Фамилия заказчика. Имя заказчика. Отчество заказчика. Адрес заказчика. Телефон заказчика. Код заказа. Дата заказа. Срок заказа. Количество экземпляров книги в заказе. Статус заказа.

## Выполнение

## Создать запросы:

- Список книг, изданных в текущем году и относящихся к категории «Базы данных».

```
select DISTINCT "Book".book_number, "Book".name, "Book".write_year, "Book".publish_year
from "Book"
```

where (select "category code" from "Category" where "name" = 'Базы данных')

```
in (select "category_code" from "Book_category" where "Book".book_number =
"Book_category".book_number)
```

and "publish\_year" = EXTRACT(YEAR FROM CURRENT\_DATE);

Query

Query History

```
1 select DISTINCT "Book".book_number, "Book".name, "Book".write_year, "Book".publish_year
2 from "Book"
3 where (select "category_code" from "Category" where "name" = 'Базы данных')
4 in (select "category_code" from "Book_category" where "Book".book_number = "Book_category".book_number)
5 and "publish_year" = EXTRACT(YEAR FROM CURRENT_DATE);
6
```

Data Output

Messages

Notifications

≡

📄

▼

📋

▼

🗑️

🗄️

⬇️

📈

	book_number [PK] integer	name character varying (50)	write_year integer	publish_year integer
1	1	Книга о волке2	1000	2023
2	2	проектирование бд	1002	2023

- Список покупателей, заказавших книг на сумму, превышающую среднюю сумму заказа за год.

Query

Query History

```

1  Select Distinct "Client".client_code,
2  "Client".first_name, "Client".last_name, "Client".father_name, "Client".address, "Client".phone
3  from "Order"
4  join "Client" on "Order".client_code = "Client".client_code
5  where "order_sum" > (select sum("order_sum") / count("order_sum") |
6                      from "Order" where "date" between CURRENT_DATE - INTERVAL '1 year' and CURRENT_DATE)
7

```

Data Output

Messages

Notifications

+

📄

▼

📋

▼

🗑️

🗄️

⬇️

📈

	client_code [PK] integer	first_name character varying (50)	last_name character varying (50)	father_name character varying (50)	address character varying (100)	phone character varying (20)
1	1	Иван	Иванов	Иванович	Питер Улица Ленина 1	+77007007070

```

Select Distinct "Client".client_code,
"Client".first_name,      "Client".last_name,      "Client".father_name,      "Client".address,
"Client".phone
from "Order"
join "Client" on "Order".client_code = "Client".client_code
where "order_sum" > (select sum("order_sum") / count("order_sum")
                    from "Order" where "date" between CURRENT_DATE -
INTERVAL '1 year' and CURRENT_DATE)

```

- Список книг, которые не заказывались в течение последних двух кварталов.

Query Query History

```

1 select * from "Book"
2 EXCEPT
3 select DISTINCT "Book".book_number, "Book".name, "Book".write_year, "Book".publish_year
4 from "Order"
5 join "Order_creation" on "Order".order_code = "Order_creation".order_code
6 join "Circulation" on "Order_creation".circulation_code = "Circulation".circulation_code
7 join "Edition" ON "Circulation".isbn_code = "Edition".isbn_code
8 join "Book" on "Edition".book_number = "Book".book_number
9 where "Order".date between CURRENT_DATE - INTERVAL '6 months' and CURRENT_DATE;
10

```

Data Output Messages Notifications

	book_number integer	name character varying (50)	write_year integer	publish_year integer
1	2	проектирование бд	1002	2002

```

select * from "Book"
EXCEPT
select      DISTINCT      "Book".book_number,      "Book".name,      "Book".write_year,
"Book".publish_year
from "Order"
join "Order_creation" on "Order".order_code = "Order_creation".order_code
join "Circulation" on "Order_creation".circulation_code = "Circulation".circulation_code
join "Edition" ON "Circulation".isbn_code = "Edition".isbn_code
join "Book" on "Edition".book_number = "Book".book_number
where "Order".date between CURRENT_DATE - INTERVAL '6 months' and
CURRENT_DATE;

```

- Список авторов, не написавших ни одной книги, относящейся к категории “Языки программирования”.

Query

Query History

```
1 select "Author".author_code, "Author".first_name, "Author".last_name, "Author".father_name, "Author".email from "Author"
2 except
3 select Distinct "Author".author_code, "Author".first_name, "Author".last_name, "Author".father_name, "Author".email
4 from "Author"
5 join "Book_author" on "Author".author_code = "Book_author".author_code
6 join "Book" on "Book".book_number = "Book_author".book_number
7 join "Book_category" on "Book".book_number = "Book_category".book_number
8 join "Category" on "Category".category_code = "Book_category".category_code
9 where "Category".name = 'Языки программирования';
```

Data Output

Messages

Notifications

	author_code integer	first_name character varying (50)	last_name character varying (50)	father_name character varying (50)	email character varying (100)
1	3	Влад	Владов	[null]	test@test.test

```

select      "Author".author_code,      "Author".first_name,      "Author".last_name,
"Author".father_name, "Author".email from "Author"

except

select  Distinct  "Author".author_code,  "Author".first_name,  "Author".last_name,
"Author".father_name, "Author".email

from "Author"

join "Book_author" on "Author".author_code = "Book_author".author_code
join "Book" on "Book".book_number = "Book_author".book_number
join "Book_category" on "Book".book_number = "Book_category".book_number
join "Category" on "Category".category_code = "Book_category".category_code
where "Category".name = 'Языки программирования';

```

- Список книг, в названиях которых содержится слово “проектирование” и которые присутствуют на базе в количестве, превышающем 50 экземпляров.

Query










Query History

```
1 select * from "Book" as "B" where "B".name like '%проектирование%' and
2 (select sum("Circulation".made_count) - sum("Circulation".left_count) as "count"
3 from "Book"
4 join "Edition" on "Book".book_number = "Edition".book_number
5 join "Circulation" on "Edition".isbn_code = "Circulation".isbn_code
6 where "Book".book_number = "B".book_number) > 50;|
7
```

Data Output

Messages

Notifications



	book_number [PK] integer	name character varying (50)	write_year integer	publish_year integer
1	2	проектирование бд	1002	2002

```

select * from "Book" as "B" where "B".name like '%проектирование%' and
(select sum("Circulation".made_count) - sum("Circulation".left_count) as "count"

```

```

from "Book"
join "Edition" on "Book".book_number = "Edition".book_number
join "Circulation" on "Edition".isbn_code = "Circulation".isbn_code
where "Book".book_number = "B".book_number) > 50;

```

- Покупателя, сделавшего заказ на максимальную сумму за последний месяц



The screenshot shows a database query editor with a 'Query' tab selected. The query is as follows:

```

1 Select Distinct "Client".client_code,
2 "Client".first_name, "Client".last_name, "Client".father_name, "Client".address, "Client".phone
3 from "Order"
4 join "Client" on "Order".client_code = "Client".client_code
5 where "order_sum" = (select max("order_sum") from "Order" where
6 "Order".date between CURRENT_DATE - INTERVAL '1 month' and CURRENT_DATE)
7 and "date" between CURRENT_DATE - INTERVAL '1 month' and CURRENT_DATE;
8

```

Below the query, the 'Data Output' tab is selected, showing a table with 6 columns: client\_code [PK] integer, first\_name character varying (50), last\_name character varying (50), father\_name character varying (50), address character varying (100), and phone character varying (20). The table contains one row of data:

client_code [PK] integer	first_name character varying (50)	last_name character varying (50)	father_name character varying (50)	address character varying (100)	phone character varying (20)
2	Петров	Петр	[null]	Москва	+78005553535

```

Select Distinct "Client".client_code,
"Client".first_name,      "Client".last_name,      "Client".father_name,      "Client".address,
"Client".phone
from "Order"
join "Client" on "Order".client_code = "Client".client_code
where "order_sum" = (select max("order_sum") from "Order" where
                        "Order".date between CURRENT_DATE - INTERVAL '1
month' and CURRENT_DATE)
and "date" between CURRENT_DATE - INTERVAL '1 month' and CURRENT_DATE;

```

- Список книг, не попавших ни в один из заказов в течение последнего года.

Query

Query History

1

select \* from "Book"

2

EXCEPT

3

select DISTINCT "Book".book\_number, "Book".name, "Book".write\_year, "Book".publish\_year

4

from "Order"

5

join "Order\_creation" on "Order".order\_code = "Order\_creation".order\_code

6

join "Circulation" on "Order\_creation".circulation\_code = "Circulation".circulation\_code

7

join "Edition" ON "Circulation".isbn\_code = "Edition".isbn\_code

8

join "Book" on "Edition".book\_number = "Book".book\_number

9

where "Order".date between CURRENT\_DATE - INTERVAL '1 year' and CURRENT\_DATE;

10

Data Output

Messages

Notifications

	book_number integer	name character varying (50)	write_year integer	publish_year integer
1	2	проектирование бд	1002	2002

select \* from "Book"

EXCEPT

select       DISTINCT       "Book".book\_number,       "Book".name,       "Book".write\_year,  
"Book".publish\_year

from "Order"

join "Order\_creation" on "Order".order\_code = "Order\_creation".order\_code

join "Circulation" on "Order\_creation".circulation\_code = "Circulation".circulation\_code

join "Edition" ON "Circulation".isbn\_code = "Edition".isbn\_code

join "Book" on "Edition".book\_number = "Book".book\_number

where   "Order".date   between   CURRENT\_DATE   -   INTERVAL   '1 year'   and  
CURRENT\_DATE;

## Создать представление

- содержащее сведения о количестве заказанных экземпляров каждой книги, изданной в текущем году;

Query	Query History
<pre> 1 create or replace view book_ordered_count as select "Book".book_number, "Book".name, 2 "Book".write_year, "Book".publish_year, sum("Order_creation".count) as "count" from "Book" 3 join "Edition" on "Book".book_number = "Edition".book_number 4 join "Circulation" on "Edition".isbn_code = "Circulation".isbn_code 5 join "Order_creation" on "Order_creation".circulation_code = "Circulation".circulation_code 6 where "publish_year" = EXTRACT(YEAR FROM CURRENT_DATE) 7 group by "Book".book_number; </pre>	

create or replace view book\_ordered\_count as select "Book".book\_number, "Book".name,



"Book".write\_year, "Book".publish\_year, sum("Order\_creation".count) as "count" from  
"Book"

join "Edition" on "Book".book\_number = "Edition".book\_number

join "Circulation" on "Edition".isbn\_code = "Circulation".isbn\_code

join "Order\_creation" on "Order\_creation".circulation\_code = "Circulation".circulation\_code

where "publish\_year" = EXTRACT(YEAR FROM CURRENT\_DATE)

group by "Book".book\_number;

Query

Query History

1

select

\*

from

book\_ordered\_count;

Data Output

Messages

Notifications

≡+

	<div>book_number</div> <div>integer</div> <div></div>	<div>name</div> <div>character varying (50)</div> <div></div>	<div>write_year</div> <div>integer</div> <div></div>	<div>publish_year</div> <div>integer</div> <div></div>	<div>count</div> <div>bigint</div> <div></div>
1	1	Книга о волке2	1000	2023	700
2	2	проектирование бд	1002	2023	100

- количество заказов по покупателям за последний год.

Query	Query History
1	create or replace view client_orders_count as select "C".client_code, "C".first_name, "C".last_name, "C".father_name, "C".phone,
2	(select count("Order".order_code) from "Client"
3	join "Order" on "Client".client_code = "Order".client_code
4	where "Order".date between CURRENT_DATE - INTERVAL '1 year' and CURRENT_DATE and "Client".client_code = "C".client_code
5	group by "Client".client_code) as "count"
6	from "Client" as "C" where (select count("Order".order_code) from "Client"
7	join "Order" on "Client".client_code = "Order".client_code
8	where "Order".date between CURRENT_DATE - INTERVAL '1 year' and CURRENT_DATE and "Client".client_code = "C".client_code
9	group by "Client".client_code) is not null
10	union
11	select "C".client_code, "C".first_name, "C".last_name, "C".father_name, "C".phone, '0'
12	from "Client" as "C" where (select count("Order".order_code) from "Client"
13	join "Order" on "Client".client_code = "Order".client_code
14	where "Order".date between CURRENT_DATE - INTERVAL '1 year' and CURRENT_DATE and "Client".client_code = "C".client_code
15	group by "Client".client_code) is null;
16	
Data Output	Messages
CREATE VIEW	
Query returned successfully in 21 msec.	

create or replace view client\_orders\_count as select "C".client\_code, "C".first\_name,  
"C".last\_name, "C".father\_name, "C".phone,

(select count("Order".order\_code) from "Client"

join "Order" on "Client".client\_code = "Order".client\_code

```

where "Order".date between CURRENT_DATE - INTERVAL '1 year' and
CURRENT_DATE and "Client".client_code = "C".client_code
group by "Client".client_code) as "count"
from "Client" as "C" where (select count("Order".order_code) from "Client"
join "Order" on "Client".client_code = "Order".client_code
where "Order".date between CURRENT_DATE - INTERVAL '1 year' and
CURRENT_DATE and "Client".client_code = "C".client_code
group by "Client".client_code) is not null
union
select "C".client_code, "C".first_name, "C".last_name, "C".father_name, "C".phone, '0'
from "Client" as "C" where (select count("Order".order_code) from "Client"
join "Order" on "Client".client_code = "Order".client_code
where "Order".date between CURRENT_DATE - INTERVAL '1 year' and
CURRENT_DATE and "Client".client_code = "C".client_code
group by "Client".client_code) is null;

```

Query

Query History

1

select \* from client\_orders\_count

Data Output

Messages

Notifications

	client_code integer	first_name character varying (50)	last_name character varying (50)	father_name character varying (50)	phone character varying (20)	count bigint
1	2	Петров	Петр	[null]	+78005553535	1
2	1	Иван	Иванов	Иванович	+77007007070	0

### 3 запроса INSERT UPDATE DELETE

1 запрос вставляет заявку на покупку издания с минимальным остатком с 2022-02-02

insert into "Order\_creation" (order\_creation\_id, order\_code, circulation\_code, count)

Values (DEFAULT, 1, 1, (select min("left\_count") from "Circulation" where "date" > '2022-02-02'));

Query	Query History	
1	insert into "Order_creation" (order_creation_id, order_code, circulation_code, count)	
2	Values (DEFAULT, 1, 1, (select min("left_count") from "Circulation" where "date" > '2022-02-02'));	
Data Output	Messages	Notifications
INSERT 0 1		
Query returned successfully in 19 msec.		

## 2 Увеличивает стоимость заказов для всех заказчиков с именем Иван

Query	Query History
<pre>1 update 2 "Order" 3 set "order_sum" = "order_sum" + 1000 4 where "client_code" in (select "Client".client_code from "Client" where "first_name" = 'Иван');</pre>	
Data Output	Messages
UPDATE 1	
Query returned successfully in 20 msec.	

## 3 удаляет все записи заказов у которых количество равно минимальному остатку на складе с 2022-02-02

delete from "Order\_creation" where "count" =  
(select min(left\_count) from "Circulation" where date > '2022-02-02');

Query	Query History	
1	delete from "Order_creation" where "count" =	
2	(select min(left_count) from "Circulation" where date > '2022-02-02');	
Data Output	Messages	Notifications
DELETE 0		
Query returned successfully in 25 msec.		

## История запросов

Query

Query History

Show queries generated internally by pgAdmin? ☒

Remove

Remove All

Today - 07.12.2023

▶ delete from "Order" where "client\_code" in (select "Client".c...

17:55:36

▶ update "Order" set "order\_sum" = "order\_sum" + 1000 where "cl...

17:54:09

▶ update "Order" set "order\_sum" = "order\_sum" + 1000 where "cl...

17:54:05

▶ update "Order" set "order\_sum" = "order\_sum" + 1000 where "cl...

17:54:00

▶ insert into "Author" (author\_code, first\_name, last\_name, fat...

17:48:54

▶ insert into "Author" (author\_code, first\_name, last\_name, fat...

17:48:48

▶ insert into "Author" (author\_code, first\_name, last\_name, fat...

17:48:30

▶ select \* from client\_orders\_count

17:43:27

▶ create or replace view client\_orders\_count as select "C".clie...

17:43:01

▶ select "C".client\_code, (select count("Order".order\_code) fro...

17:42:16

▶ select "C".client\_code, (select count("Order".order\_code) fro...

17:42:08

▶ select "C".client\_code, (select count("Order".order\_code) fro...

17:42:05

Data Output

Messages

Notifications

DELETE 0

## 4 – Запросы с индексами

Получение имени, фамилии и почты автора (без индекса)

Query Query History

```
1 select "first_name", "last_name", "email" from "Author";
```

Data Output Messages Notifications

	first_name character varying (50)	last_name character varying (50)	email character varying (100)
1	Петр	Петров	yandex@yandex.ru
2	Иван	Иванов	gmail@gmail.com
3	Влад	Владов	test@test.test
4	Иван	Иванов	test@test.test

✓ Successfully run. Total query runtime: 46 msec. 4 rows affected. ✕

create index idx\_author\_email on "Author" (email); - простой

create index idx\_author\_name on "Author" (first\_name, last\_name); - составной



Результат –

✓ Successfully run. Total query runtime: 33 msec. 4 rows affected. ✕

Ln 1, Col 57

Data Output Messages Explain ✕ Notifications

Graphical Analysis Statistics


  

  
 Author

Запрос получения автора, позиции, и книги

✓ Successfully run. Total query runtime: 85 msec. 4 rows affected. ✕

Query Query History

```
1 select "Book".name, "Book_author".author_position, "Author".first_name
2 from "Book"
3 join "Book_author" on "Book_author".book_number = "Book".book_number
4 join "Author" on "Author".author_code = "Book_author".author_code;
```

Data Output Messages Explain ✕ Notifications



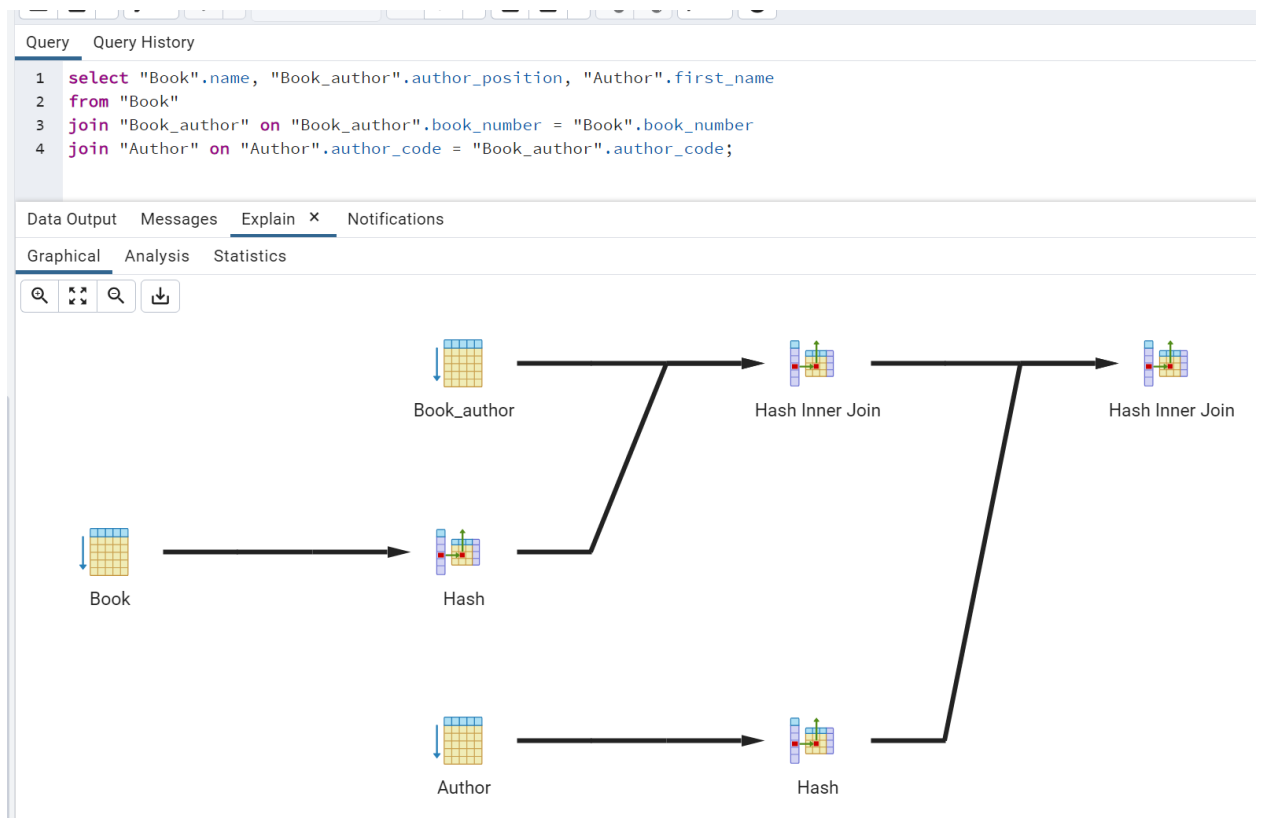
	name character varying (50) 🔒	author_position integer 🔒	first_name character varying (50) 🔒
1	Книга о волке2	1	Иван
2	Книга о волке2	2	Петр
3	проектирование бд	2	Иван
4	проектирование бд	1	Петр

create index ids\_book\_name on "Book" (name); - простой

create index ids\_book\_author on "Book\_author" (book\_number, author\_code); - составной

✓ Successfully run. Total query runtime: 29 msec. 4 rows affected. ✕

In 4 Col 67



После удаления индексов вернулось время запроса в прежнее значение

## Вывод

В данной лабораторной работе я научился строить SQL запросы, в том числе вложенные, написал свои INSERT UPDATE DELETE запросы, а также проверил работоспособность индексов и выяснил что с ними время выполнения запроса становится меньше.