

Министерство науки и высшего образования Российской Федерации
федеральное государственное автономное образовательное учреждение
высшего образования
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
ИТМО»**

Отчет

по лабораторной работе «ЗАПРОСЫ НА ВЫБОРКУ И МОДИФИКАЦИЮ ДАННЫХ,
ПРЕДСТАВЛЕНИЯ И ИНДЕКСЫ В PostgreSQL» по
дисциплине «Проектирование Баз данных»

Автор: Зайцев Кирилл Дмитриевич

Факультет: Инфокоммуникационных технологий (ИКТ)

Группа: K32402

Преподаватель: Говорова М. М.

Дата: 14.05.2023



Санкт-Петербург 2023

1 - 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.

2 - 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.

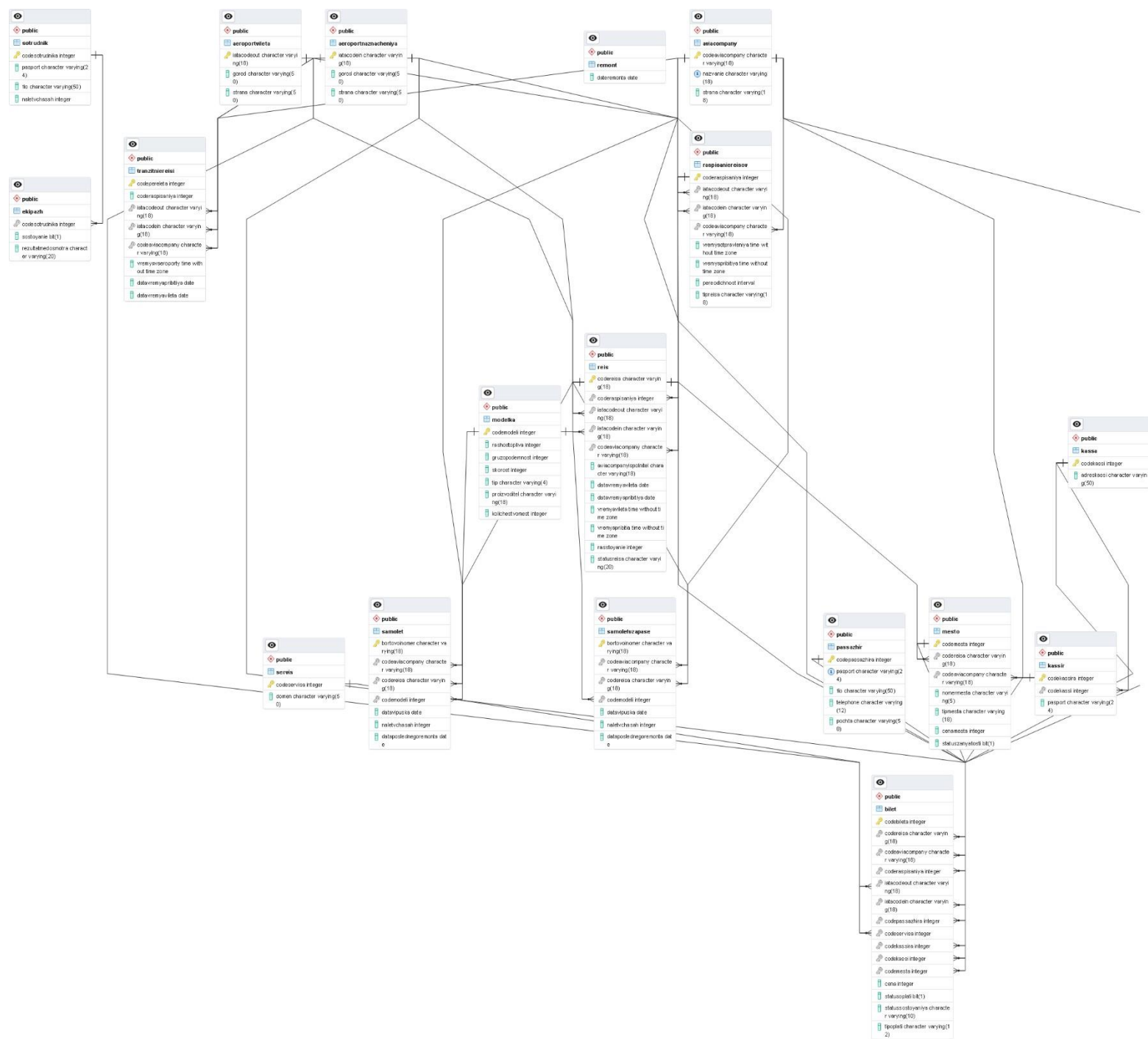
Задание 2. Создать запросы:

- Определить расчетное время полета по всем маршрутам.
- Определить расход топлива по всем маршрутам.
- Вывести данные о том, сколько свободных мест оставалось в самолетах, совершавших полет по заданному из рейсов за вчерашний день.
- Рассчитать убытки компании за счет непроданных билетов за вчерашний день.
- Определить, какой тип самолетов чаще всего летал в заданный аэропорт назначения.
- Вывести список самолетов, “возраст” которых превышает средний “возраст” самолетов этого типа.
- Определить тип самолетов, летающих во все аэропорты назначения.

Задание 3. Создать представление:

- для пассажиров авиакомпании о рейсах в Москву на ближайшую неделю;
- количество самолетов каждого типа, летавшими за последний месяц.

- Схема базы данных



– Запросы:

- Определить расчетное время полета по всем маршрутам.

Код:

```
SELECT r.codereisa, r.rasstoyanie / m.skorost AS raschetnoe_vremya_polëta
FROM reis r JOIN samolet s ON s.codereisa = r.codereisa JOIN modelka m ON
s.codeModeli = m.codeModeli WHERE rasstoyanie IS NOT NULL AND skorost
IS NOT NULL;
```

Скриншот выполнения:

	codereisa [PK] character varying (18)	raschetnoe_vremya_polëta integer
1	LK123	1
2	TK432	4
3	LK432	1
4	LK4322	1
5	LK432	1
6	LK234	1
7	LK234	1
8	LK234	1
9	LK234	1
10	LK234	1
11	LK234	1
12	LK234	1
13	LK234	1

- Определить расход топлива по всем маршрутам.

Код:

```
SELECT
    r.coderaspisaniya,
    r.iatacodeout AS вылет,
    r.iatacodein AS прилет,
    m.rashodTopлива
FROM
    raspisanieraisov r
JOIN
    reis rs ON rs.coderaspisaniya = r.coderaspisaniya
JOIN
    samolet s ON rs.bortovoiNomer = s.bortovoiNomer
JOIN
    modelka m ON s.codeModeli = m.codeModeli
Group by r.coderaspisaniya, m.rashodtopлива
```

Скриншот выполнения:

	coderaspisaniya integer	вылет character varying (18)	прилет character varying (18)	rashodtopliva integer
1	1	ULLI	FRFT	20
2	66	ULLI	FRFT	20

- Вывести данные о том, сколько свободных мест оставалось в самолетах, совершавших полет по заданному из рейсов за вчерашний день.

Код:

```
SELECT m.kolichestvomest - COUNT(b.codeBileta) AS svobodnye_mesta
FROM modelka m
JOIN samolet s ON m.codeModeli = s.codeModeli
JOIN reis r ON s.bortovoiNomer = s.bortovoiNomer
JOIN билет b ON r.codereisa = b.codereisa AND s.bortovoiNomer =
s.bortovoiNomer AND r.dataVremyaVileta = r.dataVremyaVileta
WHERE r.codereisa = 'LK234'
AND r.dataVremyaVileta = CURRENT_DATE - INTERVAL '1 day'
GROUP BY m.codeModeli, m.kolichestvomest;
```

Скриншот выполнения:

	svobodnye_mesta bigint
1	220

- Рассчитать убытки компании за счет непроданных билетов за вчерашний день.

Код:

```
SELECT
    r.codeReisa,
    SUM(
        CASE
            WHEN b.statusSostoyaniya IS NULL OR b.statusSostoyaniya = 'cancelled'
            THEN m.cenamesta
            WHEN m.statuszanyatosti IS NULL OR m.statuszanyatosti = '0'
            THEN m.cenamesta
            ELSE 0
        END
    ) AS Сумма_цен_непроданных_билетов
FROM
    reis r
LEFT JOIN
    mesto m ON r.codeReisa = m.codeReisa
LEFT JOIN
```

```

bilet b ON m.codeMesta = b.codeMesta AND b.codeReisa = r.codeReisa
WHERE
DATE(r.dataVremyaVileta) = CURRENT_DATE - INTERVAL '1 day'
GROUP BY
r.codeReisa;

```

	codereisa integer	Сумма_цен_непроданных_билетов integer
1	1488	300

- Определить, какой тип самолетов чаще всего летал в заданный аэропорт назначения.

Код:

```

WITH TypeCounts AS (
    SELECT
        r.IATACodeIN AS Аэропорт_назначения,
        m.tip AS Тип_самолета,
        COUNT(*) AS Количество_рейсов
    FROM
        reis r
        JOIN
            samolet s ON r.bortovoiNomer = s.bortovoiNomer
        JOIN
            modelka m ON s.codeModeli = m.codeModeli
    GROUP BY
        r.IATACodeIN, m.tip
),
MaxCount AS (
    SELECT
        Аэропорт_назначения,
        MAX(Количество_рейсов) AS Максимальное_количество_рейсов
    FROM
        TypeCounts
    GROUP BY
        Аэропорт_назначения
)
SELECT
    tc.Аэропорт_назначения,
    tc.Тип_самолета
FROM
    TypeCounts tc
JOIN
    MaxCount mc ON tc.Аэропорт_назначения = mc.Аэропорт_назначения

```

WHERE

tc.Количество_рейсов = mc.Максимальное_количество_рейсов;

Скриншот выполнения:

	Аэропорт_назначения character varying (18)	Тип_самолета character varying (4)
1	FRFT	pass

- Вывести список самолетов, “возраст” которых превышает средний “возраст” самолетов этого типа.

Код:

```
SELECT s.bortovoiNomer, s.codeModeli, s.dataVipuska, ((CURRENT_DATE) -  
s.dataVipuska) AS age  
FROM samolet s  
JOIN (  
SELECT codeModeli, AVG((CURRENT_DATE) - dataVipuska) AS avg_age  
FROM samolet  
GROUP BY codeModeli  
) AS t ON s.codeModeli = t.codeModeli  
WHERE (CURRENT_DATE) - s.dataVipuska > t.avg_age;
```

Скриншот выполнения:

	bortovoiNomer [PK] character varying (18)	codemodeli integer	dataVipuska date	age integer
1	GG1488	1	2020-01-01	1229
2	ST148	1	2021-01-01	863
3	PT148	1	2021-01-01	863

- Определить тип самолетов, летающих во все аэропорты назначения.

Код:

```
SELECT s.codeModeli  
FROM samolet s  
JOIN reis r ON s.codereisa = r.codereisa  
GROUP BY s.codeModeli  
HAVING COUNT(DISTINCT r.IATACodeIN) = (SELECT COUNT(*) FROM  
aeroportNaznacheniya)
```

Скриншот выполнения:

	codemodeli integer
1	1

Создать представление:

-для пассажиров авиакомпании о рейсах в Франкфурт на ближайшую неделю

Код:

```
CREATE VIEW flights_to_frankfurt_next_week AS
SELECT r.codereisa, s.bortovoiNomer , r.dataVremyaVileta ,
r.dataVremyaPribitiya
FROM samolet s
JOIN reis r ON s.bortovoinomer = r.bortovoinomer
WHERE r.IATACodeIN = 'FRFT'
AND r.dataVremyaVileta BETWEEN CURRENT_DATE AND
CURRENT_DATE + INTERVAL '7 days' AND s.codemodeli = 2;
```

Скриншоты выполнения:

CREATE VIEW	
Query returned successfully in 33 msec.	
definition text	
1	SELECT s.codemodeli, count(DISTINCT r.codereisa) AS num_flights FROM (samolet s JOIN reis r ON (((s.codereisa)::text = (r.codereisa)::text))) WHERE (r.datavremyavileta >= '2020-04-01':date) GROUP BY s.codemodeli;

Проверка:

select * from flights_to_frankfurt_next_week

	codereisa character varying (50)	bortovoinomer character varying (50)	datavremyavileta date	datavremyapribitiya date
1	LK234	BB148	2023-10-04	2023-10-04

-количество самолетов каждого типа, летавшими за последний месяц

Код:

```
CREATE VIEW num_flights_by_type AS
SELECT s.codeModeli, COUNT(DISTINCT r.codereisa) AS num_flights
FROM samolet s
JOIN reis r ON s.codereisa = r.codereisa
WHERE r.datavremyavileta >= CURRENT_DATE - INTERVAL '1 month'
GROUP BY s.codeModeli;
```

Скриншоты выполнения:

CREATE VIEW

Query returned successfully in 33 msec.

	schemaname name	viewname name	viewowner name	definition text
1	public	num_flights_by_type	postgres	SELECT s.codemodeli, count(DISTINCT r.codereisa) AS num_flights FROM (samolet s JOIN reis r ON (((s.codereisa)::text = (r.codereisa)::text))) WHERE (r.datavremyavieta >= '2020-04-01'::date) GROUP BY s.codemodeli;

Проверка:

Select * from num_flights_by_type

	codemodeli integer	num_flights bigint
1	2	1

3 запроса на модификацию данных (INSERT, UPDATE, DELETE) с использованием подзапросов.

INSERT:

Добавление в таблицу запасных самолетов, тех самолетов, у которых давно не производился техосмотр.

Код:

```
INSERT INTO samoletvzapase (bortovoiNomer , codeAviacompany , codeReisa ,  
codeModeli , dataVipuska , naletVchasah , dataPoslednegoRemonta)  
SELECT bortovoiNomer , codeAviacompany , codeReisa , codeModeli ,  
dataVipuska , naletVchasah , dataPoslednegoRemonta FROM samolet  
WHERE dataPoslednegoRemonta < '01.01.2023';
```

Скриншот таблицы до команды:

	bortovoiNomer [PK] character varying (18)	codeaviacompany character varying (18)	codereisa character varying (18)	codemodel integer	datavipuska date	naletvchasa integer	dataposlednegoremonta date
1	OK148	lufthanza1	LK234	2	2023-05-01	300	2023-05-14
2	AA148	lufthanza1	LK234	2	2018-01-01	300	2023-05-14

Скриншот после выполнения запроса:

	bortovoiNomer [PK] character varying (18)	codeaviacompany character varying (18)	codereisa character varying (18)	codemodel integer	datavipuska date	naletvchasa integer	dataposlednegoremonta date
1	OK148	lufthanza1	LK234	2	2023-05-01	300	2023-05-14
2	AA148	lufthanza1	LK234	2	2018-01-01	300	2023-05-14
3	ZV148	lufthanza1	LK234	2	2015-01-01	300	2019-01-01

DELETE: Удаляет из списка активных самолётов те, которые были отправлены на ремонт/Т.О.

Код:

```
DELETE FROM samolet
WHERE bortovoiNomer IN
(SELECT bortovoiNomer FROM samoletvzapase)
AND dataPoslednegoRemonta < '2023-01-01';
```

Скриншот до выполнения:

	bortovoiNomer [PK] character varying (18)	codeaviacompany character varying (18)	codereisa character varying (18)	codemodel integer	datavipuska date	naletvchasa integer	dataposlednegoremonta date
4	GG1488	lufthanza1	LK432	1	2020-01-01	300	2023-01-01
5	ST148	lufthanza1	TK432	1	2021-01-01	300	2023-01-01
6	PT148	lufthanza1	LK123	1	2021-01-01	300	2023-01-01
7	BG148	lufthanza1	LK234	2	2023-01-01	300	2023-01-01
8	BV148	lufthanza1	LK234	2	2023-01-01	300	2023-01-01
9	BZ148	lufthanza1	LK234	2	2023-01-01	300	2023-01-01
10	BG140	lufthanza1	LK234	2	2023-01-01	300	2023-01-01
11	BV141	lufthanza1	LK234	2	2023-01-01	300	2023-01-01
12	BZ142	lufthanza1	LK234	2	2023-01-01	300	2023-01-01
13	BG143	lufthanza1	LK234	2	2023-01-01	300	2023-01-01
14	BV144	lufthanza1	LK234	2	2023-01-01	300	2023-01-01
15	BZ145	lufthanza1	LK234	2	2023-01-01	300	2023-01-01
16	ZV148	lufthanza1	LK234	2	2015-01-01	300	2019-01-01

Скриншот после выполнения:

	bortovoiNomer [PK] character varying (18)	codeaviacompany character varying (18)	codereisa character varying (18)	codemodel integer	datavipuska date	naletvchasa integer	dataposlednegoremonta date
3	GG158	lufthanza1	LK4322	1	2023-01-01	300	2023-01-01
4	GG1488	lufthanza1	LK432	1	2020-01-01	300	2023-01-01
5	ST148	lufthanza1	TK432	1	2021-01-01	300	2023-01-01
6	PT148	lufthanza1	LK123	1	2021-01-01	300	2023-01-01
7	BG148	lufthanza1	LK234	2	2023-01-01	300	2023-01-01
8	BV148	lufthanza1	LK234	2	2023-01-01	300	2023-01-01
9	BZ148	lufthanza1	LK234	2	2023-01-01	300	2023-01-01
10	BG140	lufthanza1	LK234	2	2023-01-01	300	2023-01-01
11	BV141	lufthanza1	LK234	2	2023-01-01	300	2023-01-01
12	BZ142	lufthanza1	LK234	2	2023-01-01	300	2023-01-01
13	BG143	lufthanza1	LK234	2	2023-01-01	300	2023-01-01
14	BV144	lufthanza1	LK234	2	2023-01-01	300	2023-01-01
15	BZ145	lufthanza1	LK234	2	2023-01-01	300	2023-01-01

UPDATE:

Обновляет данные о последнем ремонте/Т.О самолета находящегося на ремонте/Т.О.

```
Код:
UPDATE samoletvzapase
SET dataPoslednegoRemonta = (
    SELECT dateremonta
    FROM remont
    WHERE
        SUBSTRING(bortovoiNomer FROM 1 FOR 2) = 'BB'
        AND codeModeli = 2
)
WHERE
    SUBSTRING(bortovoiNomer FROM 1 FOR 2) = 'BB'
    AND codeModeli = 2;
```

Скриншот до:

	bortovoiNomer [PK] character varying (18)	codeaviacompany character varying (18)	codeModeli integer	datavipuska date	naletvchasa h integer	dataposlednegoremonta date
1	BB148	lufthanza1	2	2023-01-01	300	2023-01-01

Скриншот после:

	bortovoiNomer [PK] character varying (18)	codeaviacompany character varying (18)	codeModeli integer	datavipuska date	naletvchasa h integer	dataposlednegoremonta date
1	BB148	lufthanza1	2	2023-01-01	300	2023-09-27

ИНДЕКСЫ:

Запрос без индекса:

1 Select * from samolet WHERE(codereisa = 'LK234')

Data Output Messages Graph Visualiser Explain Notifications

	bortovoiNomer [PK] character varying (18)	codeaviacompany character varying (18)	codereisa character varying (18)	codeModeli integer	datavipuska date	naletvchasa h integer	data poslednegoremonta date
1	BB148	lufthanza1	LK234	2	2023-01-01	300	2023-01-01
2	BG148	lufthanza1	LK234	2	2023-01-01	300	2023-01-01
3	BV148	lufthanza1	LK234	2	2023-01-01	300	2023-01-01
4	BZ148	lufthanza1	LK234	2	2023-01-01	300	2023-01-01
5	BG140	lufthanza1	LK234	2	2023-01-01	300	2023-01-01
6	BV141	lufthanza1	LK234	2	2023-01-01	300	2023-01-01
7	BZ142	lufthanza1	LK234	2	2023-01-01	300	2023-01-01
8	BG143	lufthanza1	LK234	2	2023-01-01	300	2023-01-01
9	BV144	lufthanza1	LK234	2	2023-01-01	300	2023-01-01
10	BZ145	lufthanza1	LK234	2	2023-01-01	300	2023-01-01

План запроса:

Query

Query History

1

EXPLAIN Select * from samolet WHERE(codereisa = 'LK234')

Data Output

Messages

Graph Visualiser

×

Explain

×

Notifications

≡+

📄

▼

📋

🗑️

🗄️

⬇️

📈

	QUERY PLAN	🔒
	text	
1	Seq Scan on samolet (cost=0.00..1.19 rows=1 width=178)	
2	Filter: ((codereisa)::text = 'LK234'::text)	

Код индекса:

Create index idx_codereisa on samolet (codereisa) Скриншот
выполнения запроса:

Query

Query History

1

Select * from samolet WHERE(codereisa = 'LK234')

Data Output

Messages

Graph Visualiser

Explain

Notifications

	bortovnomer [PK] character varying (18)	codeavlacompany character varying (18)	codereisa character varying (18)	codemodel integer	datavipuska date	naletvchasa integer	data poslednego remonta date
1	BB148	lufthanza1	LK234	2	2023-01-01	300	2023-01-01
2	BG148	lufthanza1	LK234	2	2023-01-01	300	2023-01-01
3	BV148	lufthanza1	LK234	2	2023-01-01	300	2023-01-01
4	BZ148	lufthanza1	LK234	2	2023-01-01	300	2023-01-01
5	BG140	lufthanza1	LK234	2	2023-01-01	300	2023-01-01
6	BV141	lufthanza1	LK234	2	2023-01-01	300	2023-01-01
7	BZ142	lufthanza1	LK234	2	2023-01-01	300	2023-01-01
8	BG143	lufthanza1	LK234	2	2023-01-01	300	2023-01-01
9	BV144	lufthanza1	LK234	2	2023-01-01	300	2023-01-01
10	BZ145	lufthanza1	LK234	2	2023-01-01	300	2023-01-01

План запроса:

Query Query History	
1 EXPLAIN Select * from samolet WHERE (codereisa = 'LK234')	
Data Output Messages Graph Visualiser Explain Notifications	
<div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> </div>	
	<div> <div>QUERY PLAN</div> <div>text</div> </div>
1	Seq Scan on samolet (cost=0.00..1.19 rows=1 width=178)
2	Filter: ((codereisa)::text = 'LK234')::text)

Запрос без индекса:

Query
Query History

```

1 SELECT r.codereisa, r.dataVremyaVileta, m.tip, s.codeModeli AS ModelSamoleta
2 FROM reis r
3 JOIN samolet s ON r.codereisa = s.codereisa
4 JOIN modelka m ON s.codeModeli = m.codeModeli
5 WHERE r.IATACodeOUT = 'ULLI' AND m.tip = 'gruz';

```

Data Output
Messages
Graph Visualiser
Explain
Notifications

	codereisa character varying (18)	datavremyavileta date	tip character varying (4)	modelsamoleta integer
1	LK432	2023-04-04	gruz	1
2	LK4322	2023-04-04	gruz	1
3	LK432	2023-04-04	gruz	1
4	TK432	2023-04-04	gruz	1

План запроса:

Query
Query History

```

1 Explain SELECT r.codereisa, r.dataVremyaVileta, m.tip, s.codeModeli AS ModelSamoleta
2 FROM reis r
3 JOIN samolet s ON r.codereisa = s.codereisa
4 JOIN modelka m ON s.codeModeli = m.codeModeli
5 WHERE r.IATACodeOUT = 'ULLI' AND m.tip = 'gruz';

```

Data Output
Messages
Graph Visualiser
Explain
Notifications

	QUERY PLAN text
1	Nested Loop (cost=1.24..9.97 rows=1 width=82)
2	-> Hash Join (cost=1.09..2.29 rows=2 width=62)
3	Hash Cond: ((s.codereisa)::text = (r.codereisa)::text)
4	-> Seq Scan on samolet s (cost=0.00..1.15 rows=15 width=58)
5	-> Hash (cost=1.07..1.07 rows=1 width=58)
6	-> Seq Scan on reis r (cost=0.00..1.07 rows=1 width=58)
7	Filter: (((iatacodeout)::text = 'ULLI')::text)
8	-> Index Scan using modelka_pkey on modelka m (cost=0.15..3.10 rows=1 width=24)
9	Index Cond: (codemodeli = s.codemodeli)
10	Filter: (((tip)::text = 'gruz')::text)

Создание индекса:

Create index idx_codereisa_codeModeli on samolet (codereisa, codeModeli)

Запрос с индексом:

1 SELECT r.codereisa, r.dataVremyaVileta, m.tip, s.codeModeli AS ModelSamoleta

2 FROM reis r

3 JOIN samolet s ON r.codereisa = s.codereisa

4 JOIN modelka m ON s.codeModeli = m.codeModeli

5 WHERE r.IATACodeOUT = 'ULLI' AND m.tip = 'gruz';

Data Output Messages Graph Visualiser Explain Notifications

	codereisa character varying (18)	dataVremyaVileta date	tip character varying (4)	modelsamoleta integer
1	LK432	2023-04-04	gruz	1
2	LK4322	2023-04-04	gruz	1
3	LK432	2023-04-04	gruz	1
4	TK432	2023-04-04	gruz	1

План запроса:

Query Query History

1 Explain SELECT r.codereisa, r.dataVremyaVileta, m.tip, s.codeModeli AS ModelSamoleta

2 FROM reis r

3 JOIN samolet s ON r.codereisa = s.codereisa

4 JOIN modelka m ON s.codeModeli = m.codeModeli

5 WHERE r.IATACodeOUT = 'ULLI' AND m.tip = 'gruz';

Data Output Messages Graph Visualiser Explain Notifications

QUERY PLAN

text

1	Nested Loop (cost=1.24..9.97 rows=1 width=82)
2	-> Hash Join (cost=1.09..2.29 rows=2 width=62)
3	Hash Cond: ((s.codereisa)::text = (r.codereisa)::text)
4	-> Seq Scan on samolet s (cost=0.00..1.15 rows=15 width=58)
5	-> Hash (cost=1.07..1.07 rows=1 width=58)
6	-> Seq Scan on reis r (cost=0.00..1.07 rows=1 width=58)
7	Filter: ((iatacodeout)::text = 'ULLI')::text)
8	-> Index Scan using modelka_pkey on modelka m (cost=0.15..3.10 rows=1 width=24)
9	Index Cond: (codemodeli = s.codemodeli)
10	Filter: ((tip)::text = 'gruz')::text)

После выполнения запросов:

-Как в первом, так и во втором случае выполнение запроса с индексом заняло больше времени, чем без него. Вероятнее всего по той причине, что таблица заполнена НЕ большим объёмом данных.

Выводы:

В результате выполнения данной работы я овладел практическими навыками работы с базой данных PostgreSQL. Я научился создавать представления и делать запросы на выборку данных, а также использовать подзапросы при модификации данных. Кроме того, я познакомился с индексами и научился создавать и использовать простые и составные индексы для ускорения выполнения запросов. В целом, выполнение данной работы помогло мне лучше понять, как работать с PostgreSQL и как оптимизировать запросы для более быстрого доступа к данным. Эти навыки могут быть полезными при разработке приложений, которые используют базы данных PostgreSQL.

Список использованных источников:

1. Документация PostgreSQL. Запросы [Электронный ресурс] // Официальный сайт PostgreSQL. 1996-2023. URL: <https://www.postgresql.org/docs/10/queries.html> (дата обращения: 02.03.2023).
2. Документация pgAdmin 4 PostgreSQL. Query Tool [Электронный ресурс] // Официальный сайт pgAdmin . URL: https://www.pgadmin.org/docs/pgadmin4/latest/query_tool.html (дата обращения: 02.03.2023).