

**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»

Обучающиеся Меньшенин Евгений

Факультет прикладной информатики

Группа К3241

Направление подготовки 09.03.03 Прикладная информатика

Образовательная программа Мобильные и сетевые технологии 2023

Преподаватель Говорова Марина Михайловна

Санкт-Петербург
2025

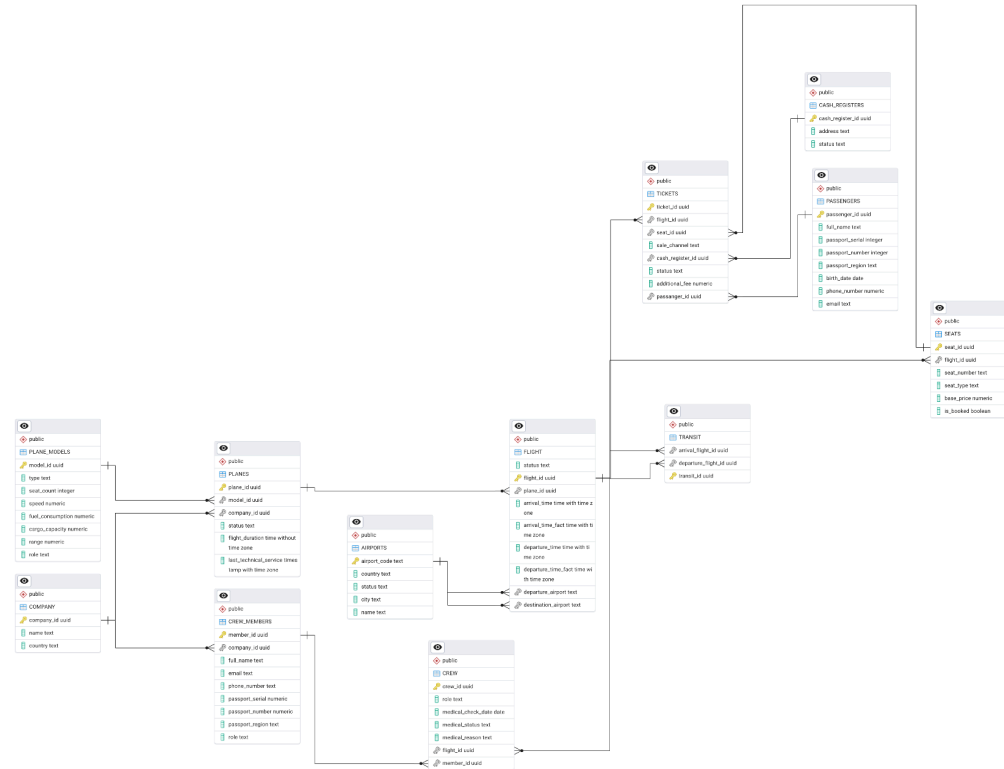
Цель работы: овладеть практическими навыками создания представлений и запросов на выборку данных к базе данных PostgreSQL, использования подзапросов при модификации данных и индексов.

Практическое задание.

1. Создать запросы и представления на выборку данных к базе данных PostgreSQL (согласно индивидуальному заданию лабораторной работы №2, 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.

Ход работы

Наименование базы данных: airports



Запросы к базе данных

-- 1 Определить расчетное время полета по всем маршрутам.

SELECT

flight_id,

departure_airport,

destination_airport,

departure_time,

arrival_time,

(arrival_time - departure_time) AS calculated_flight_time

FROM

public."FLIGHT";

Data Output Messages Graph Visualiser X Notifications						
	flight_id [PK] uuid	departure_airport text	destination_airport text	departure_time timestamp with time zone	arrival_time timestamp with time zone	calculated_flight_time interval
1	6cae20f2-796d-49fc-b33b-3c4021e2e1b0	SVO	JFK	2025-05-10 08:00:00+03	2025-05-10 13:50:00+03	05:50:00
2	ac74fa13-594d-4304-91a0-e3cdeed1c408	JFK	CDG	2025-05-11 09:10:00+03	2025-05-11 15:15:00+03	06:05:00
3	10000000-0000-0000-0000-000000000001	SVO	JFK	2025-06-01 12:00:00+03	2025-06-01 20:00:00+03	08:00:00
4	10000000-0000-0000-0000-000000000002	JFK	DME	2025-06-02 07:00:00+03	2025-06-02 13:00:00+03	06:00:00
5	10000000-0000-0000-0000-000000000003	DME	LHR	2025-06-03 11:00:00+03	2025-06-03 18:00:00+03	07:00:00
6	10000000-0000-0000-0000-000000000004	LHR	CDG	2025-06-04 09:30:00+03	2025-06-04 15:00:00+03	05:30:00
7	10000000-0000-0000-0000-000000000005	CDG	SVO	2025-06-05 13:20:00+03	2025-06-05 19:00:00+03	05:40:00
8	20000000-0000-0000-0000-000000000001	SVO	JFK	2025-06-02 13:00:00+03	2025-06-02 22:00:00+03	09:00:00
9	20000000-0000-0000-0000-000000000002	JFK	DME	2025-06-03 08:00:00+03	2025-06-03 16:00:00+03	08:00:00
10	20000000-0000-0000-0000-000000000003	DME	LHR	2025-06-04 10:00:00+03	2025-06-04 19:00:00+03	09:00:00
11	20000000-0000-0000-0000-000000000004	LHR	CDG	2025-06-05 09:30:00+03	2025-06-05 15:00:00+03	05:30:00
12	20000000-0000-0000-0000-000000000005	CDG	SVO	2025-06-06 11:20:00+03	2025-06-06 18:00:00+03	06:40:00

-- 2 Определить расход топлива по всем маршрутам.

SELECT

f.flight_id,

f.departure_airport,

f.destination_airport,

(f.arrival_time - f.departure_time) AS flight_time,

pm.fuel_consumption * EXTRACT(EPOCH FROM (f.arrival_time -
f.departure_time)) / 3600 AS estimated_fuel_usage

FROM

public."FLIGHT" f

JOIN public."PLANES" p ON f.plane_id = p.plane_id

JOIN public."PLANE_MODELS" pm ON p.model_id = pm.model_id;

29 FROM

Data Output Messages Graph Visualiser X Notifications

SQL

	flight_id [PK] uuid	departure_airport text	destination_airport text	flight_time interval	estimated_fuel_usage numeric
1	6cae20f2-796d-49fc-b33b-3c4021e2e1b0	SVO	JFK	05:50:00	43.7500000000000000
2	ac74fa13-594d-4304-91a0-e3cdeed1c408	JFK	CDG	06:05:00	32.8500000000000000
3	10000000-0000-0000-0000-000000000001	SVO	JFK	08:00:00	60.0000000000000000
4	10000000-0000-0000-0000-000000000002	JFK	DME	06:00:00	45.0000000000000000
5	10000000-0000-0000-0000-000000000003	DME	LHR	07:00:00	52.5000000000000000
6	10000000-0000-0000-0000-000000000004	LHR	CDG	05:30:00	41.2500000000000000
7	10000000-0000-0000-0000-000000000005	CDG	SVO	05:40:00	42.5000000000000000
8	20000000-0000-0000-0000-000000000001	SVO	JFK	09:00:00	48.6000000000000000
9	20000000-0000-0000-0000-000000000002	JFK	DME	08:00:00	43.2000000000000000
10	20000000-0000-0000-0000-000000000003	DME	LHR	09:00:00	48.6000000000000000
11	20000000-0000-0000-0000-000000000004	LHR	CDG	05:30:00	29.7000000000000000
12	20000000-0000-0000-0000-000000000005	CDG	SVO	06:40:00	36.0000000000000000

-- 3 Вывести данные о том, сколько свободных мест оставалось в самолетах, совершавших полет по заданному рейсу за вчерашний день

```
SELECT
    f.flight_id,
    COUNT(s.seat_id) FILTER (WHERE s.is_booked = false) AS free_seats
FROM
    public."FLIGHT" f
JOIN public."SEATS" s ON f.flight_id = s.flight_id
WHERE
    f.departure_airport = 'SVO'
    AND f.destination_airport = 'JFK'
    AND f.departure_time::date = CURRENT_DATE - INTERVAL '1 day'
GROUP BY
    f.flight_id;
```












42 `SUM(s.base_price + t.additional_fee) AS lost_ir`

Data Output Messages Graph Visualiser X Notifications

	flight_id [PK] uuid	free_seats bigint
1	10000000-0000-0000-0000-0000000000...	0
2	20000000-0000-0000-0000-0000000000...	0
3	6cae20f2-796d-49fc-b33b-3c4021e2e1b0	2

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

```
SELECT
  f.flight_id,
  SUM(s.base_price + t.additional_fee) AS lost_income
FROM
  public."FLIGHT" f
JOIN public."SEATS" s ON f.flight_id = s.flight_id
JOIN public."TICKETS" t ON f.flight_id = t.flight_id
WHERE
  f.arrival_time::date = CURRENT_DATE - INTERVAL '1 day'
  AND s.is_booked = false
GROUP BY
  f.flight_id;
```

Data Output Messages Graph Visualiser X Notifications		
         SQL		
	flight_id [PK] uuid 	lost_income numeric 
1	6cae20f2-796d-49fc-b33b-3c4021e2e1b0	31600


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


```
SELECT
    pm.type,
    COUNT(*) AS flights_count
FROM
    public."FLIGHT" f
JOIN public."PLANES" p ON f.plane_id = p.plane_id
JOIN public."PLANE_MODELS" pm ON p.model_id = pm.model_id
WHERE
    f.destination_airport = 'JFK'
GROUP BY
    pm.type
ORDER BY
    flights_count DESC
LIMIT 1;
```


Data Output


Messages


Graph Visual


















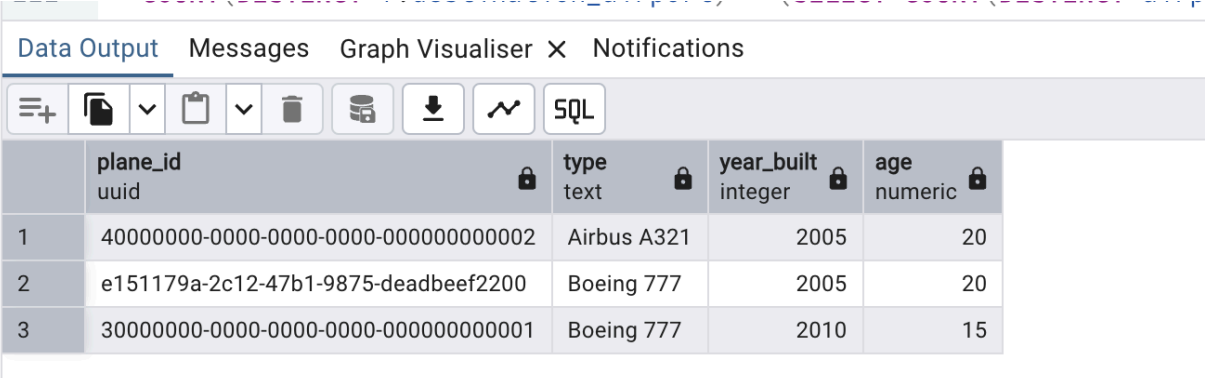




	<div>type</div> <div>text</div> <div></div>	<div>flights_count</div> <div>bigint</div> <div></div>
1	Boeing 777	2

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

```
WITH plane_ages AS (  
    SELECT p.plane_id, pm.type, p.year_built,  
           EXTRACT(YEAR FROM CURRENT_DATE) - p.year_built AS age  
    FROM public."PLANES" p  
    JOIN public."PLANE_MODELS" pm ON p.model_id = pm.model_id  
)  
avg_ages AS (  
    SELECT type, AVG(age) AS avg_age  
    FROM plane_ages  
    GROUP BY type  
)  
SELECT pa.plane_id, pa.type, pa.year_built, pa.age  
FROM plane_ages pa  
JOIN avg_ages aa ON pa.type = aa.type  
WHERE  
    pa.age > aa.avg_age  
ORDER BY  
    pa.type, pa.age DESC;
```



	plane_id uuid	type text	year_built integer	age numeric
1	40000000-0000-0000-0000-000000000002	Airbus A321	2005	20
2	e151179a-2c12-47b1-9875-deadbeef2200	Boeing 777	2005	20
3	30000000-0000-0000-0000-000000000001	Boeing 777	2010	15

-- 7 Определить тип самолетов, летающих во все аэропорты назначения.

SELECT

pm.type

FROM

public."PLANES" p

JOIN public."PLANE_MODELS" pm ON p.model_id = pm.model_id











JOIN public."FLIGHT" f ON p.plane_id = f.plane_id

GROUP BY

pm.type

HAVING

COUNT(DISTINCT f.destination_airport) = (SELECT COUNT(DISTINCT airport_code) FROM public."AIRPORTS");

112	
Data Output Messages Graph Visualiser X	
         SQL	
	type text 
1	Airbus A321
2	Boeing 777

Представления

-- 1 для пассажиров авиакомпании о рейсах в Москву на ближайшую неделю

CREATE OR REPLACE VIEW public.flights_to_moscow_next_week AS

SELECT

f.flight_id,

f.status,

f.departure_airport,

dep.city AS departure_city,

f.destination_airport,

arr.city AS destination_city,

f.departure_time,

f.arrival_time

FROM

public."FLIGHT" f

JOIN public."AIRPORTS" arr ON f.destination_airport = arr.airport_code

JOIN public."AIRPORTS" dep ON f.departure_airport = dep.airport_code

WHERE

arr.city ILIKE 'Moscow'

AND f.departure_time::date BETWEEN CURRENT_DATE AND
CURRENT_DATE + INTERVAL '7 day'

ORDER BY

f.departure_time;

Notifications

	flight_id uuid	status text	departure_airport text	departure_city text	destination_airport text	destination_city text	departure_time timestamp with time zone	arrival_time timestamp with time zone
1	aaaaaaaa-bbbb-cccc-dddd-eeeeeeeeeeee	Scheduled	LHR	London	SVO	Moscow	2025-05-06 16:00:00+03	2025-05-06 18:20:00+03

-- 2 количество самолетов каждого типа, летавшими за последний месяц

CREATE OR REPLACE VIEW public.planes_per_type_last_month AS

SELECT

pm.type AS plane_type,

COUNT(DISTINCT p.plane_id) AS planes_count

FROM

public."FLIGHT" f

JOIN public."PLANES" p ON f.plane_id = p.plane_id

JOIN public."PLANE_MODELS" pm ON p.model_id = pm.model_id

WHERE

f.departure_time >= (CURRENT_DATE - INTERVAL '1 month')

AND f.departure_time <= NOW()

GROUP BY

pm.type

ORDER BY

pm.type;

Data Output Messages Notifications			
<div><div><div><div></div></div><div><div></div></div><div><div></div></div><div><div></div></div><div><div></div></div><div><div></div></div><div><div></div></div><div><div></div></div><div><div></div></div><div><div></div></div><div><div></div></div><div><div></div></div></div><div>SQL</div></div>			
	plane_type text	planes_count bigint	
1	Airbus A320		1

Запросы на модификацию данных

1. INSERT

Data Output Messages Notifications											
	member_id [PK] uuid	company_id uuid	full_name text	email text	phone_number text	passport_serial numeric	passport_number numeric	passport_region text	role text		
1	b104ec73-09fa-4fa0-a487-e4627b7c1e...	e1169c5a-068a-4ffa-802f-bf6896c3c47c	Ivan Petrov	ivan.petrov@aerostudent.com	+79161234567	1234	567890	Moscow	Pilot		
2	6177bfd3-4272-41e7-92e0-1079c5555...	b5f9b1d9-1111-4222-9333-444455556...	Pierre Martin	pierr.martin@airfrance.fr	+33123456789	6412	9988771	Paris	Pilot		
3	80037fad-0340-4015-ab3f-a2c6f0d76fbb	e3434343-cafe-4b3d-b111-121ac3507777	Elena Sidorova	elena.sidorova@s7.ru	+74996661234	1002	87654322	Novosibirsk	FlightAttendant		

```
INSERT INTO "CREW_MEMBERS"(member_id, company_id, full_name, email,
phone_number, passport_serial, passport_number, passport_region, role)
VALUES (
    uuid_generate_v4(),
    (SELECT company_id FROM "COMPANY" WHERE name = 'AeroStudent'
LIMIT 1),
    'Pavel Ivanov',
    'pavel.ivanov@aerostudent.com',
    '+79261234567',
    1444,
    1234567,
    'Moscow',
    'Pilot'
);
```

Data Output Messages Notifications											
	member_id [PK] uuid	company_id uuid	full_name text	email text	phone_number text	passport_serial numeric	passport_number numeric	passport_region text	role text		
1	b104ec73-09fa-4fa0-a487-e4627b7c1e...	e1169c5a-068a-4ffa-802f-bf6896c3c47c	Ivan Petrov	ivan.petrov@aerostudent.com	+79161234567	1234	567890	Moscow	Pilot		
2	6177bfd3-4272-41e7-92e0-1079c5555...	b5f9b1d9-1111-4222-9333-444455556...	Pierre Martin	pierr.martin@airfrance.fr	+33123456789	6412	9988771	Paris	Pilot		
3	80037fad-0340-4015-ab3f-a2c6f0d76fbb	e3434343-cafe-4b3d-b111-121ac3507...	Elena Sidorova	elena.sidorova@s7.ru	+74996661234	1002	87654322	Novosibirsk	FlightAttendant		
4	ff3288b9-06a3-4138-bd8e-ad607f51f001	aa2ff6be-e37a-455e-adc3-227dca12caaf	Pavel Ivanov	pavel.ivanov@aerostudent.com	+79261234567	1444	1234567	Moscow	Pilot		

2. UPDATE

Data Output

Messages

Notifications

SQL

	cash_register_id [PK] uuid	address text	status text
1	a430db59-ca6e-4eae-acf3-7a77733cfe09	Moscow Airport Terminal B	Open
2	163304b5-3f0b-48a1-9534-a7cc61e944...	New York Terminal 4	InUse
3	fa1d8dba-60fa-4f7d-bc3a-8817f09d2791	London Heathrow Terminal 3	Open
4	2e5d7bee-43bb-450d-88f5-18e2c563c5b8	Paris CDG Terminal 2E	Closed

UPDATE "CASH_REGISTERS"

SET status = 'UnderMaintenance'

WHERE address ILIKE ANY (

ARRAY(

```
SELECT '%' || city || '%'
```

FROM "AIRPORTS"

WHERE country = 'Russia'

)

$$);$$

Data Output

Messages

Notifications

SQL

	cash_register_id [PK] uuid	address text	status text
1	163304b5-3f0b-48a1-9534-a7cc61e944...	New York Terminal 4	InUse
2	fa1d8dba-60fa-4f7d-bc3a-8817f09d2791	London Heathrow Terminal 3	Open
3	2e5d7bee-43bb-450d-88f5-18e2c563c5...	Paris CDG Terminal 2E	Closed
4	a430db59-ca6e-4eae-acf3-7a77733cfe09	Moscow Airport Terminal B	UnderMaintenance

3. DELETE











Data Output Messages Notifications										
Showing rows: 1 to 11 Page No:										
	ticket_id [PK] uuid	flight_id uuid	seat_id uuid	sale_channel text	cash_register_id uuid	status text	additional_fee numeric	passenger_id uuid		
1	94c7245e-e63c-4380-8499-60b2eb24e4da	6cae20f2-796d-49fc-b33b-3c4021e2e1b0	974aa408-12c0-4e8e-b039-91b3d10ab1da	Online	a430db59-cafe-4eae-acf3-7a77733cfe09	Booked	0	858d93c1-5f17-40d8-bacd-aabd23d8f1b5		
2	80000000-0000-0000-0000-000000000001	10000000-0000-0000-0000-000000000001	60000000-0000-0000-0000-000000000001	Online	a430db59-cafe-4eae-acf3-7a77733cfe09	Booked	0	858d93c1-5f17-40d8-bacd-aabd23d8f1b5		
3	80000000-0000-0000-0000-000000000002	10000000-0000-0000-0000-000000000002	60000000-0000-0000-0000-000000000002	Online	a430db59-cafe-4eae-acf3-7a77733cfe09	Booked	0	110daaa8-c0de-4eee-af09-979dda82b9...		
4	80000000-0000-0000-0000-000000000003	10000000-0000-0000-0000-000000000003	60000000-0000-0000-0000-000000000003	Online	a430db59-cafe-4eae-acf3-7a77733cfe09	Booked	0	feed1f26-3f55-49c2-bbb1-db2b781654ef		
5	80000000-0000-0000-0000-000000000004	10000000-0000-0000-0000-000000000004	60000000-0000-0000-0000-000000000004	Online	a430db59-cafe-4eae-acf3-7a77733cfe09	Booked	0	858d93c1-5f17-40d8-bacd-aabd23d8f1b5		
6	80000000-0000-0000-0000-000000000005	10000000-0000-0000-0000-000000000005	60000000-0000-0000-0000-000000000005	Online	a430db59-cafe-4eae-acf3-7a77733cfe09	Booked	0	110daaa8-c0de-4eee-af09-979dda82b9...		
7	80000000-0000-0000-0000-000000000006	20000000-0000-0000-0000-000000000001	60000000-0000-0000-0000-000000000006	Online	163304b5-3f0b-48a1-9534-a7cc61e944ab	Booked	0	feed1f26-3f55-49c2-bbb1-db2b781654ef		
8	80000000-0000-0000-0000-000000000007	20000000-0000-0000-0000-000000000002	60000000-0000-0000-0000-000000000007	Online	163304b5-3f0b-48a1-9534-a7cc61e944ab	Booked	0	858d93c1-5f17-40d8-bacd-aabd23d8f1b5		
9	80000000-0000-0000-0000-000000000008	20000000-0000-0000-0000-000000000003	60000000-0000-0000-0000-000000000008	Online	163304b5-3f0b-48a1-9534-a7cc61e944ab	Booked	0	110daaa8-c0de-4eee-af09-979dda82b9...		
10	80000000-0000-0000-0000-000000000009	20000000-0000-0000-0000-000000000004	60000000-0000-0000-0000-000000000009	Online	163304b5-3f0b-48a1-9534-a7cc61e944ab	Booked	0	feed1f26-3f55-49c2-bbb1-db2b781654ef		
11	80000000-0000-0000-0000-000000000010	20000000-0000-0000-0000-000000000005	60000000-0000-0000-0000-000000000010	Online	163304b5-3f0b-48a1-9534-a7cc61e944ab	Booked	0	858d93c1-5f17-40d8-bacd-aabd23d8f1b5		

```
DELETE FROM "TICKETS" t
WHERE passanger_id IN (
    SELECT p.passenger_id
    FROM "PASSENGERS" p
    WHERE p.birth_date < (CURRENT_DATE - INTERVAL '40 years')
)
AND flight_id IN (
    SELECT f.flight_id
    FROM "FLIGHT" f
    WHERE f.departure_airport IN (
        SELECT airport_code FROM "AIRPORTS"
        WHERE city = 'Moscow'
    )
);
```











Data Output Messages Notifications										
Showing rows: 1										
	ticket_id [PK] uuid	flight_id uuid	seat_id uuid	sale_channel text	cash_register_id uuid	status text	additional_fee numeric	passenger_id uuid		
1	94c7245e-e63c-4380-8499-60b2eb24e4da	6cae20f2-796d-49fc-b33b-3c4021e2e1b0	974aa408-12c0-4e8e-b039-91b3d10ab1da	Online	a430db59-cafe-4eae-acf3-7a77733cfe09	Booked	0	858d93c1-5f17-40d8-bacd-aabd23d8f1b5		
2	80000000-0000-0000-0000-000000000001	10000000-0000-0000-0000-000000000001	60000000-0000-0000-0000-000000000001	Online	a430db59-cafe-4eae-acf3-7a77733cfe09	Booked	0	858d93c1-5f17-40d8-bacd-aabd23d8f1b5		
3	80000000-0000-0000-0000-000000000002	10000000-0000-0000-0000-000000000002	60000000-0000-0000-0000-000000000002	Online	a430db59-cafe-4eae-acf3-7a77733cfe09	Booked	0	110daaa8-c0de-4eee-af09-979dda82b9...		
4	80000000-0000-0000-0000-000000000004	10000000-0000-0000-0000-000000000004	60000000-0000-0000-0000-000000000004	Online	a430db59-cafe-4eae-acf3-7a77733cfe09	Booked	0	858d93c1-5f17-40d8-bacd-aabd23d8f1b5		
5	80000000-0000-0000-0000-000000000005	10000000-0000-0000-0000-000000000005	60000000-0000-0000-0000-000000000005	Online	a430db59-cafe-4eae-acf3-7a77733cfe09	Booked	0	110daaa8-c0de-4eee-af09-979dda82b9...		
6	80000000-0000-0000-0000-000000000007	20000000-0000-0000-0000-000000000002	60000000-0000-0000-0000-000000000007	Online	163304b5-3f0b-48a1-9534-a7cc61e944ab	Booked	0	858d93c1-5f17-40d8-bacd-aabd23d8f1b5		
7	80000000-0000-0000-0000-000000000008	20000000-0000-0000-0000-000000000003	60000000-0000-0000-0000-000000000008	Online	163304b5-3f0b-48a1-9534-a7cc61e944ab	Booked	0	110daaa8-c0de-4eee-af09-979dda82b9...		
8	80000000-0000-0000-0000-000000000009	20000000-0000-0000-0000-000000000004	60000000-0000-0000-0000-000000000009	Online	163304b5-3f0b-48a1-9534-a7cc61e944ab	Booked	0	feed1f26-3f55-49c2-bbb1-db2b781654ef		
9	80000000-0000-0000-0000-000000000010	20000000-0000-0000-0000-000000000005	60000000-0000-0000-0000-000000000010	Online	163304b5-3f0b-48a1-9534-a7cc61e944ab	Booked	0	858d93c1-5f17-40d8-bacd-aabd23d8f1b5		

Индексы

```
EXPLAIN ANALYZE SELECT * FROM public."PLANES"  
WHERE company_id = 'e3434343-cafe-4b3d-b111-121ac3507777';
```

Data Output		Messages	Notifications
        			
	QUERY PLAN text		
1	Seq Scan on "PLANES" (cost=0.00..150.70 rows=5 width=100) (actual time=0.019..3.098 rows=1695 loops=...		
2	Filter: (company_id = 'e3434343-cafe-4b3d-b111-121ac3507777'::uuid)		
3	Rows Removed by Filter: 8313		
4	Planning Time: 0.318 ms		
5	Execution Time: 3.265 ms		

```
CREATE INDEX idx_planes_company_id ON public."PLANES"(company_id);
```

Data Output		Messages	Notifications
        			
	QUERY PLAN text		
1	Bitmap Heap Scan on "PLANES" (cost=4.32..21.51 rows=5 width=100) (actual time=0.111..0.634 rows=1695 loops=1)		
2	Recheck Cond: (company_id = 'e3434343-cafe-4b3d-b111-121ac3507777'::uuid)		
3	Heap Blocks: exact=137		
4	-> Bitmap Index Scan on idx_planes_company_id (cost=0.00..4.31 rows=5 width=0) (actual time=0.072..0.072 rows=1695 loop...		
5	Index Cond: (company_id = 'e3434343-cafe-4b3d-b111-121ac3507777'::uuid)		
6	Planning Time: 0.109 ms		
7	Execution Time: 0.707 ms		


```
EXPLAIN ANALYZE SELECT * FROM public."SEATS"
WHERE flight_id = '6cae20f2-796d-49fc-b33b-3c4021e2e1b0' AND seat_type = 'Economy'
AND is_booked = FALSE;
```

Data Output Messages Notifications	
QUERY PLAN	
text	
1	Seq Scan on "SEATS" (cost=0.00..254.19 rows=3546 width=51) (actual time=0.014..1.277 rows=3517 loops=1)
2	Filter: ((NOT is_booked) AND (flight_id = '6cae20f2-796d-49fc-b33b-3c4021e2e1b0'::uuid) AND (seat_type = 'Economy'::text))
3	Rows Removed by Filter: 6496
4	Planning Time: 0.247 ms
5	Execution Time: 1.403 ms

```
CREATE INDEX idx_seats_flightid_type_booked ON public."SEATS"(flight_id, seat_type,
is_booked);
```

Data Output Messages Notifications	
QUERY PLAN	
text	
1	Bitmap Heap Scan on "SEATS" (cost=61.50..218.69 rows=3546 width=51) (actual time=0.124..0.634 rows=3517 loops=1)
2	Recheck Cond: ((flight_id = '6cae20f2-796d-49fc-b33b-3c4021e2e1b0'::uuid) AND (seat_type = 'Economy'::text) AND (NOT is_booked))
3	Heap Blocks: exact=104
4	-> Bitmap Index Scan on idx_seats_flightid_type_booked (cost=0.00..60.61 rows=3546 width=0) (actual time=0.102..0.103 rows=3517 loop...)
5	Index Cond: ((flight_id = '6cae20f2-796d-49fc-b33b-3c4021e2e1b0'::uuid) AND (seat_type = 'Economy'::text) AND (is_booked = false))
6	Planning Time: 0.104 ms
7	Execution Time: 0.762 ms

Выводы

Наполнил базу данных данными, сделал запросы и представления по варианту, создал, удалил и изменил существующие данные с использованием подзапросов. Создал простой и составной индексы и проверил эффективность их работы.