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: Петухов С.А.

Факультет: ИКТ

Группа: К3139

Преподаватель: Говорова М.М.

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:

Создать запросы и представления на выборку данных к базе данных PostgreSQL (согласно индивидуальному заданию, часть 2 и 3).

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

Изучить графическое представление запросов и просмотреть историю запросов.

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.

Выполнение:

Запросы

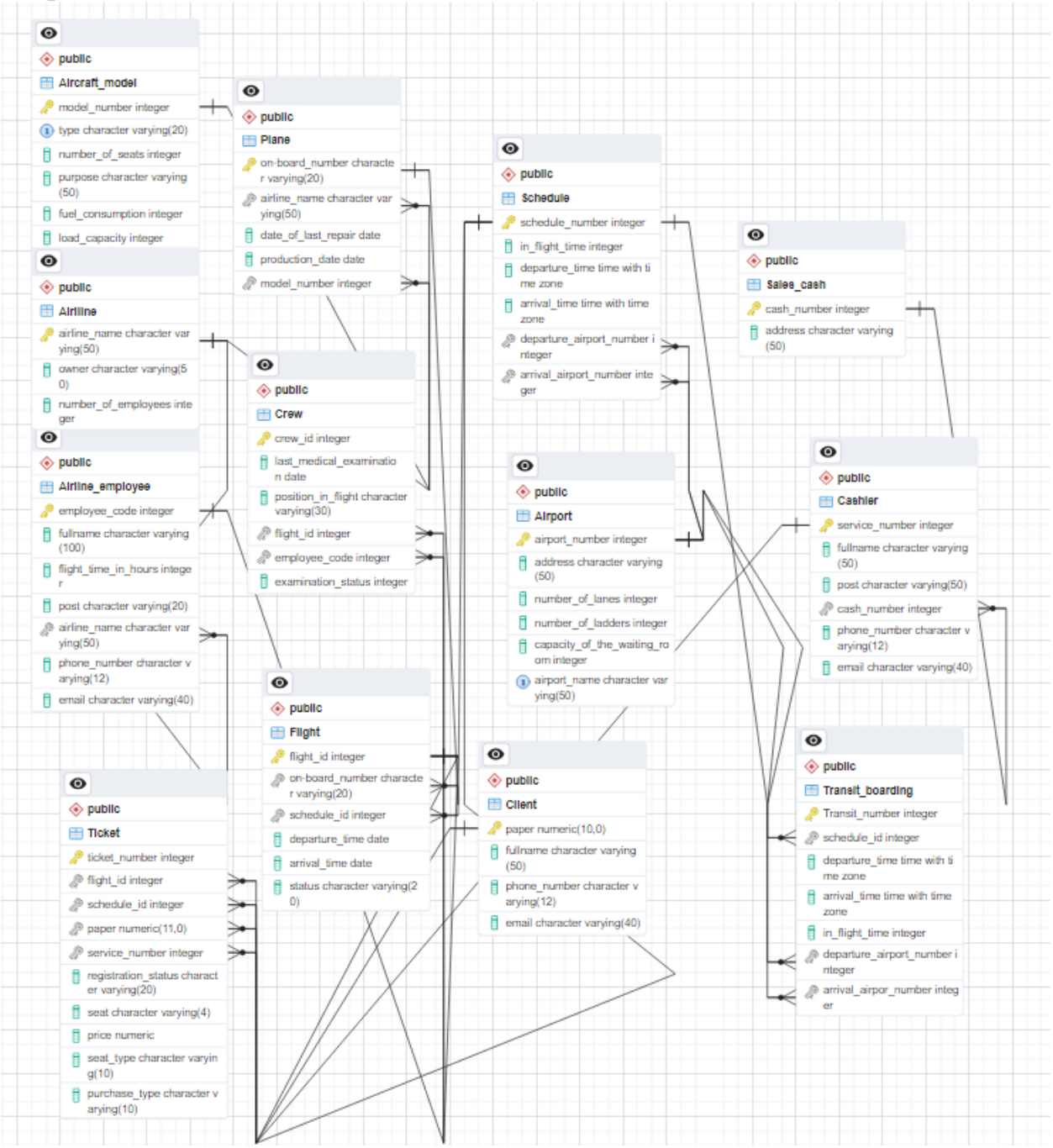


Рисунок 1 – Схема базы данных

Запросы

1)Определить расчетное время полета по всем маршрутам.

```
SELECT schedule_number,
       in_flight_time
FROM public."Schedule";
```

Query Query History

```
SELECT schedule_number,
       in_flight_time
FROM public."Schedule";

```

Data Output Messages Notifications

	schedule_number [PK] integer	in_flight_time integer
1	4	5
2	5	24
3	6	7
4	7	4
5	8	6
6	12	6
7	13	3
8	39	13
9	40	3

Successfully run. Total query runtime: 149 msec. 58 rows affected.

Total rows: 58 of 58 Query complete 00:00:00.149 Ln 4, Col 1

2)Определить расход топлива по всем маршрутам.

```
SELECT
    flight_id,
    f."on-board_number",
    f.schedule_id,
    a.fuel_consumption * s.in_flight_time AS fuel_consumption_on_flight
FROM
    public."Flight" f
JOIN
    public."Plane" p ON f."on-board_number" = p."on-board_number"
JOIN
    public."Aircraft_model" a ON p.model_number = a.model_number
JOIN
    public."Schedule" s ON f.schedule_id = s.schedule_number;
```

Query | Query History

```
SELECT
    flight_id,
    f."on-board_number",
    f.schedule_id,
    a.fuel_consumption * s.in_flight_time AS fuel_consumption_on_flight
FROM
    public."Flight" f
JOIN
    public."Plane" p ON f."on-board_number" = p."on-board_number"
JOIN
    public."Aircraft_model" a ON p.model_number = a.model_number
JOIN
    public."Schedule" s ON f.schedule_id = s.schedule_number;
```

Data Output | Messages | Notifications

	flight_id [PK] integer	on-board_number character varying (20)	schedule_id integer	fuel_consumption_on_flight integer
1	1	ПУ-205	4	75
2	2	НЕ-1488	8	126
3	3	ХК-504	7	40
4	4	ПД-3534	5	360
5	5	НЕ-1488	4	105
6	8	ХК-504	61	150
7	9	ПУ-3894	52	200
8	10	ПУ-1933	58	320
9	11	ПУ-2349	82	260

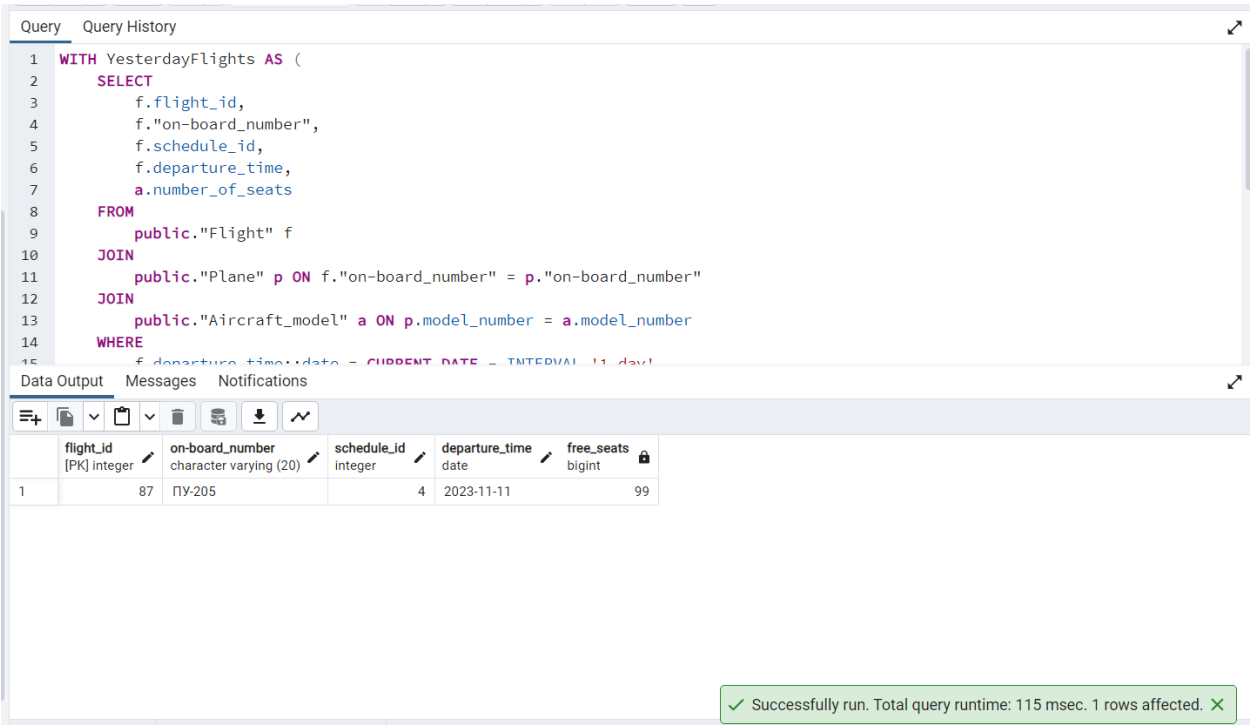
Successfully run. Total query runtime: 191 msec. 83 rows affected.

Total rows: 83 of 83 | Query complete 00:00:00.191 | Ln 13, Col 62

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

```
WITH YesterdayFlights AS (
    SELECT
        f.flight_id,
        f."on-board_number",
        f.schedule_id,
        f.departure_time,
        a.number_of_seats
    FROM
        public."Flight" f
    JOIN
        public."Plane" p ON f."on-board_number" = p."on-board_number"
    JOIN
        public."Aircraft_model" a ON p.model_number = a.model_number
    WHERE
        f.departure_time::date = CURRENT_DATE - INTERVAL '1 day'
)

SELECT
    yf.flight_id,
    yf."on-board_number",
    yf.schedule_id,
    yf.departure_time,
    yf.number_of_seats - COALESCE(COUNT(t.ticket_number), 0) AS free_seats
FROM
    YesterdayFlights yf
LEFT JOIN
    public."Ticket" t ON yf.flight_id = t.flight_id
WHERE
    t.seat IS NOT NULL
GROUP BY
    yf.flight_id, yf."on-board_number", yf.schedule_id, yf.departure_time,
yf.number_of_seats;
```



4)Рассчитать убытки компании за счет непроданных билетов за вчерашний день.

```
WITH RevenuePerFlight AS (
    SELECT
        f.flight_id,
       f."on-board_number",
        f.schedule_id,
        AVG(t.price) AS avg_ticket_price,
        COUNT(t.ticket_number) AS total_sold_tickets
    FROM
        public."Flight" f
    LEFT JOIN
        public."Ticket" t ON f.flight_id = t.flight_id
    WHERE
        t.seat IS NOT NULL
    GROUP BY
        f.flight_id, f."on-board_number", f.schedule_id
)

SELECT
    a.airline_name,
    ROUND(COALESCE(SUM(rapf.avg_ticket_price * (rapf.total_sold_tickets -
am.number_of_seats)), 0)*-1) AS total_losses
FROM
    public."Airlline" a
LEFT JOIN
    public."Plane" p ON a.airline_name = p.airline_name
LEFT JOIN
    public."Flight" f ON p."on-board_number" = f."on-board_number"
LEFT JOIN
    public."Aircraft_model" am ON p.model_number = am.model_number
LEFT JOIN
    RevenuePerFlight rapf ON f.flight_id = rapf.flight_id
GROUP BY
    a.airline_name;
```

Query Query History

```
WITH RevenuePerFlight AS (
    SELECT
        f.flight_id,
        f."on-board_number",
        f.schedule_id,
        AVG(t.price) AS avg_ticket_price,
        COUNT(t.ticket_number) AS total_sold_tickets
    FROM
        public."Flight" f
    LEFT JOIN
        public."Ticket" t ON f.flight_id = t.flight_id
    WHERE
        t.seat IS NOT NULL
    GROUP BY
```

Data Output Messages Notifications

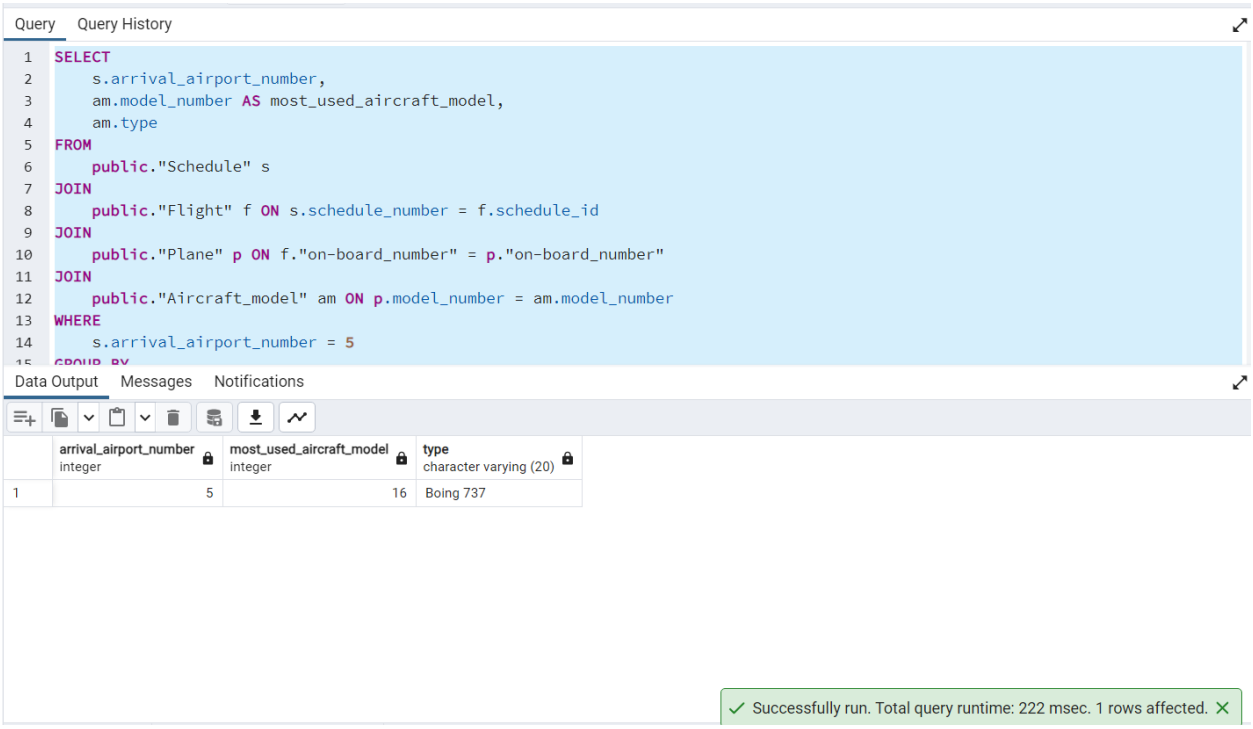
	airline_name [PK] character varying (50)	total_losses numeric
1	Россия	3113786.9999999999810000
2	Победа	8110371.0000000000000000
3	Аэрофлот	12285228.0000000000000000
4	S7 AIRLINES	19498718.0000000000000000
5	Уральские авиалинии	9898644.0000000000000000

Successfully run. Total query runtime: 102 msec. 5 rows affected.

Total rows: 5 of 5 Query complete 00:00:00.102 Ln 21, Col 6

5)Определить, какой тип самолетов чаще всего летал в заданный аэропорт назначения(у нас в аэропорт с номером 5).

```
WITH AircraftCounts AS (
    SELECT
        s.arrival_airport_number,
        am.model_number AS aircraft_model,
        am.type AS aircraft_type,
        COUNT(f."on-board_number") AS flight_count,
        RANK() OVER (ORDER BY COUNT(f."on-board_number") DESC) AS rank
    FROM
        public."Schedule" s
    JOIN
        public."Flight" f ON s.schedule_number = f.schedule_id
    JOIN
        public."Plane" p ON f."on-board_number" = p."on-board_number"
    JOIN
        public."Aircraft_model" am ON p.model_number = am.model_number
    WHERE
        s.arrival_airport_number = 5
    GROUP BY
        s.arrival_airport_number, am.model_number, am.type
)
SELECT
    arrival_airport_number,
    aircraft_model AS most_used_aircraft_model,
    aircraft_type AS most_used_aircraft_type
FROM
    AircraftCounts
WHERE
    rank = 1;
```



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

```
SELECT
    p."on-board_number",
    p.production_date,
    EXTRACT(YEAR FROM AGE(NOW(), p.production_date)) AS aircraft_age,
    ROUND(avg_model_age.avg_age) AS average_model_age
FROM
    public."Plane" p
JOIN (
    SELECT
        model_number,
        AVG(EXTRACT(YEAR FROM AGE(NOW(), production_date))) AS avg_age
    FROM
        public."Plane"
    GROUP BY
        model_number
) AS avg_model_age ON p.model_number = avg_model_age.model_number
WHERE
    EXTRACT(YEAR FROM AGE(NOW(), p.production_date)) > avg_model_age.avg_age;
```

Query | Query History

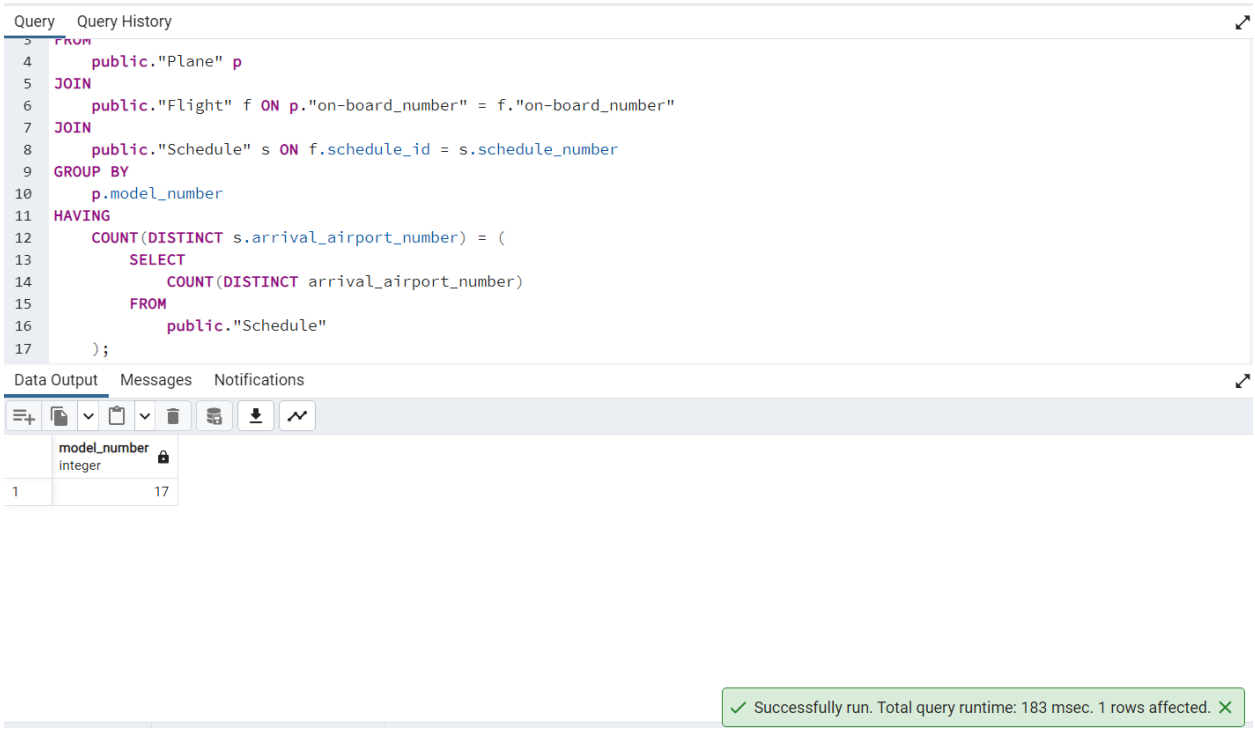
```
    avg_model_age.avg_age AS average_model_age
FROM
    public."Plane" p
JOIN (
    SELECT
        model_number,
        AVG(EXTRACT(YEAR FROM AGE(NOW(), production_date))) AS avg_age
    FROM
        public."Plane"
    GROUP BY
        model_number
) AS avg_model_age ON p.model_number = avg_model_age.model_number
WHERE
    EXTRACT(YEAR FROM AGE(NOW(), p.production_date)) > avg_model_age.avg_age;
```

Data Output | Messages | Notifications

	on-board_number [PK] character varying (20)	production_date date	aircraft_age numeric	average_model_age numeric
1	ПД-3534	2000-10-10	23	21.5000000000000000
2	ПУ-3907	1994-09-11	29	17.2400000000000000
3	ПУ-3467	1996-12-09	26	17.2400000000000000
4	ПУ-1590	1993-04-07	30	17.2400000000000000
5	ПУ-260	2003-05-05	20	17.2400000000000000
6	ПУ-1667	2003-09-20	20	17.2400000000000000
7	ПУ-4644	2005-09-05	18	17.2400000000000000
8	ПУ-2490	1997-08-02	26	17.2400000000000000
9	ПУ-1266	2004-05-03	19	17.2400000000000000

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

```
WITH AircraftDestinationCounts AS (
    SELECT
        am.model_number,
        am.type,
        COUNT(DISTINCT s.arrival_airport_number) AS destination_count
    FROM
        public."Aircraft_model" am
    JOIN
        public."Plane" p ON am.model_number = p.model_number
    JOIN
        public."Flight" f ON p."on-board_number" = f."on-board_number"
    JOIN
        public."Schedule" s ON f.schedule_id = s.schedule_number
    JOIN
        public."Airport" a ON s.arrival_airport_number = a.airport_code
    GROUP BY
        am.model_number, am.type
)
SELECT
    model_number,
    type
FROM
    AircraftDestinationCounts
WHERE
    destination_count = (
        SELECT COUNT(DISTINCT airport_code)
        FROM public."Airport"
    );
```



Представления

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

```
CREATE VIEW public."MoscowFlightsView" AS
SELECT
    f.flight_id,
    f."on-board_number",
    f.departure_time,
    f.arrival_time,
    s_departure.airport_name AS departure_airport,
    s_arrival.airport_name AS arrival_airport,
    p.airline_name
FROM
    public."Flight" f
JOIN
    public."Schedule" s ON f.schedule_id = s.schedule_number
JOIN
    public."Plane" p ON f."on-board_number" = p."on-board_number"
JOIN
    public."Airport" s_departure ON s.departure_airport_number =
s_departure.airport_number
JOIN
    public."Airport" s_arrival ON s.arrival_airport_number =
s_arrival.airport_number
WHERE
    s_arrival.address SIMILAR TO '%Москва%'
    AND f.departure_time BETWEEN CURRENT_TIMESTAMP AND CURRENT_TIMESTAMP +
INTERVAL '7 days'
ORDER BY
    f.departure_time;
```

Query Query History

```
1  SELECT * FROM public."MoscowFlightsView"
2
```

Data Output Messages Notifications

	flight_id integer	on-board_number character varying (20)	departure_time date	arrival_time date	departure_airport character varying (50)	arrival_airport character varying (50)	airline_name character varying (50)
1	92	ПУ-205	2023-11-16	2023-11-16	Ленинградский аэропорт	Пулково	S7 AIRLINES

2)количество самолетов каждого типа, летавшими за последний месяц.

```
CREATE VIEW public."AircraftCountByModelLastMonth" AS
SELECT
    am.type AS aircraft_type,
    COUNT(DISTINCT p."on-board_number") AS aircraft_count
FROM
    public."Plane" p
JOIN
    public."Flight" f ON p."on-board_number" = f."on-board_number"
JOIN
    public."Aircraft_model" am ON p.model_number = am.model_number
WHERE
    f.departure_time BETWEEN CURRENT_DATE - INTERVAL '1 month' AND
CURRENT_DATE
GROUP BY
    am.type;
```

Query Query History

```
1  SELECT * FROM public."AircraftCountByModelLastMonth"
2
```

Data Output Messages Notifications

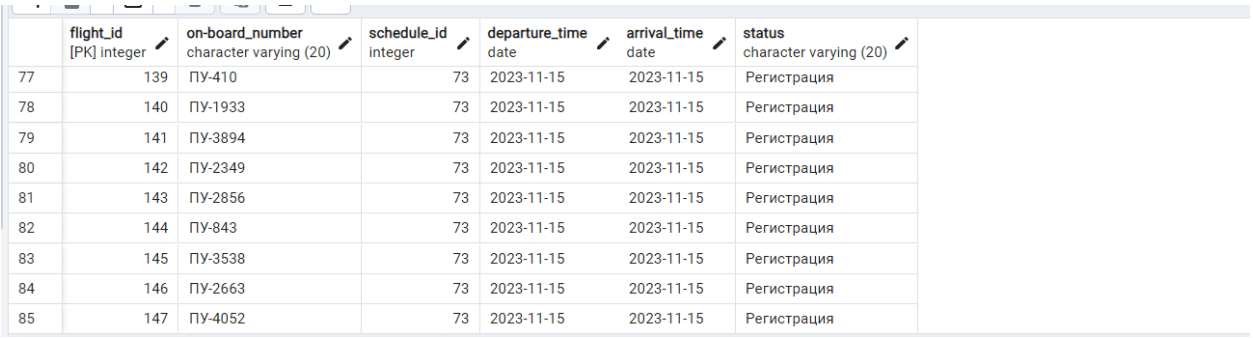
	aircraft_type character varying (20)	aircraft_count bigint
1	Airbus A319	3
2	Heinkel He 111	2
3	АН-225	1

Запросы на модификацию данных.

INSERT

Создать рейсы в аэропорт «Внуково» для авиакомпании «Аэрофлот»

```
INSERT INTO public."Flight" ("on-board_number", schedule_id, departure_time,
arrival_time, status)
SELECT
    p."on-board_number",
    s.schedule_number,
    '2023-11-15 08:00:00',
    '2023-11-15 10:00:00',
    'Регистрация'
FROM
    public."Schedule" s
JOIN
    public."Airport" a ON s.departure_airport_number = a.airport_number
JOIN
    public."Plane" p ON p.airline_name = 'Аэрофлот'
WHERE
    a.airport_name = 'Внуково'
ORDER BY
    p."on-board_number";
```



	flight_id [PK] integer	on-board_number character varying (20)	schedule_id integer	departure_time date	arrival_time date	status character varying (20)
77	139	ПУ-410	73	2023-11-15	2023-11-15	Регистрация
78	140	ПУ-1933	73	2023-11-15	2023-11-15	Регистрация
79	141	ПУ-3894	73	2023-11-15	2023-11-15	Регистрация
80	142	ПУ-2349	73	2023-11-15	2023-11-15	Регистрация
81	143	ПУ-2856	73	2023-11-15	2023-11-15	Регистрация
82	144	ПУ-843	73	2023-11-15	2023-11-15	Регистрация
83	145	ПУ-3538	73	2023-11-15	2023-11-15	Регистрация
84	146	ПУ-2663	73	2023-11-15	2023-11-15	Регистрация
85	147	ПУ-4052	73	2023-11-15	2023-11-15	Регистрация

DELETE

Удаление дубликатов из flight

```
DELETE FROM public."Flight" a
USING public."Flight" b
WHERE a.flight_id < b.flight_id
  AND a."on-board_number" = b."on-board_number"
  AND a.schedule_id = b.schedule_id
  AND a.departure_time = b.departure_time
  AND a.arrival_time = b.arrival_time;
```

UPDATE

Замена статуса на «Завершен» для окончившихся рейсов.

```
UPDATE public."Flight"
SET status = 'Завершён'
WHERE arrival_time < CURRENT_TIMESTAMP;
```

Индексы

CREATE INDEX idx_plane_on_board_number

ON public."Plane" ("on-board_number");

CREATE INDEX idx_plane_model_number

ON public."Plane" ("model_number");

CREATE INDEX idx_aircraft_model_model_number

ON public."Aircraft_model" ("model_number");

Без индексов

Query Query History

```
1  SELECT
2      flight_id,
3      f."on-board_number",
4      f.schedule_id,
5      a.fuel_consumption * s.in_flight_time AS fuel_consumption_on_flight
6  FROM
7      public."Flight" f
8  JOIN
9      public."Plane" p ON f."on-board_number" = p."on-board_number"
10 JOIN
11     public."Aircraft_model" a ON p.model_number = a.model_number
12 JOIN
13     public."Schedule" s ON f.schedule_id = s.schedule_number;
```

Data Output | Messages | Explain × | Notifications

```
Successfully run. Total query runtime: 119 msec.
85 rows affected.
```

С индексами

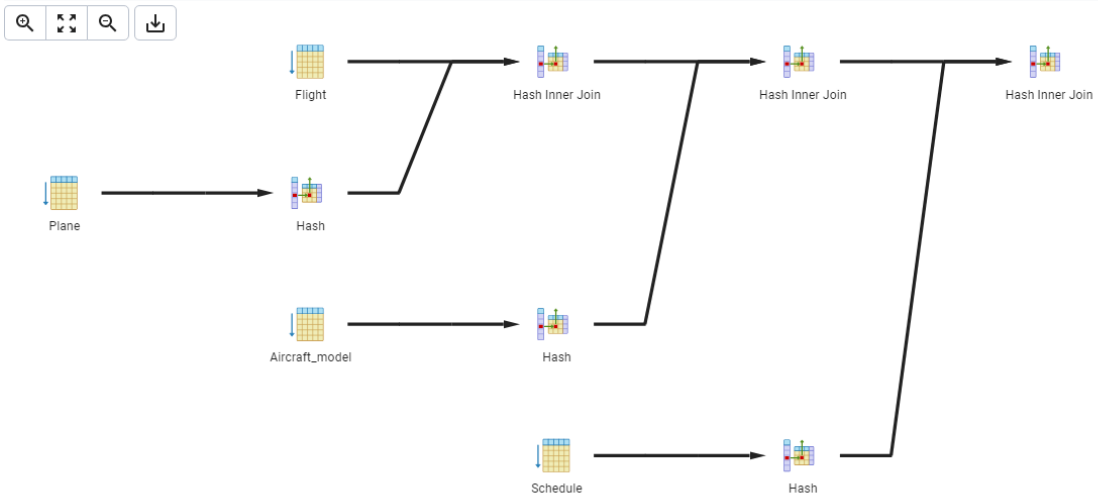
Query Query History

```
1  SELECT
2      flight_id,
3      f."on-board_number",
4      f.schedule_id,
5      a.fuel_consumption * s.in_flight_time AS fuel_consumption_on_flight
6  FROM
7      public."Flight" f
8  JOIN
9      public."Plane" p ON f."on-board_number" = p."on-board_number"
10 JOIN
11     public."Aircraft_model" a ON p.model_number = a.model_number
12 JOIN
13     public."Schedule" s ON f.schedule_id = s.schedule_number;
```

Data Output | Messages | Explain × | Notifications

```
Successfully run. Total query runtime: 77 msec.
1 rows affected.
```

	#	Node	Rows Actual	Loops
	1.	→ Hash Inner Join (rows=85 loops=1) **Hash Cond**: (f.schedule_id = s.schedule_number)	85	
●	2.	→ Hash Inner Join (rows=85 loops=1) **Hash Cond**: (p.model_number = a.model_number)	85	
	3.	→ Hash Inner Join (rows=85 loops=1) **Hash Cond**: ((f."on-board_number")::text = (p."on-board_number")::text)	85	
	4.	→ Seq Scan on Flight as f (rows=85 loops=1)	85	
	5.	→ Hash (rows=56 loops=1) **Buckets**: 1024 **Batches**: 1 **Memory Usage**: 11 kB	56	
	6.	→ Seq Scan on Plane as p (rows=56 loops=1)	56	
	7.	→ Hash (rows=9 loops=1) **Buckets**: 1024 **Batches**: 1 **Memory Usage**: 9 kB	9	
	8.	→ Seq Scan on Aircraft_model as a (rows=9 loops=1)	9	
●	9.	→ Hash (rows=58 loops=1) **Buckets**: 1024 **Batches**: 1 **Memory Usage**: 11 kB	58	
	10.	→ Seq Scan on Schedule as s (rows=58 loops=1)	58	



Составной индекс

CREATE INDEX idx_schedule_number_in_flight_time

ON public."Schedule" (schedule_number, in_flight_time);

Без индекса

Query Query History

```
1  SELECT schedule_number,
2         in_flight_time
3  FROM public."Schedule";
```

Data Output Messages Explain Notifications

```
Successfully run. Total query runtime: 181 msec.
58 rows affected.
```

С индексом

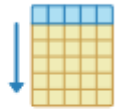
```
1  SELECT schedule_number,
2         in_flight_time
3  FROM public."Schedule";
```

Data Output Messages Explain Notifications

```
Successfully run. Total query runtime: 77 msec.
1 rows affected.
```

Graphical Analysis Statistics

#	Node	Rows Actual	Loops
1.	→ Seq Scan on Schedule as Schedule (rows=58 loops=1)	58	1



Schedule

Удаление индексов

DROP INDEX idx_plane_on_board_number;

DROP INDEX idx_plane_model_number;

DROP INDEX idx_aircraft_model_model_number;

DROP INDEX idx_schedule_number_in_flight_time;

Выводы:
В этой работе были изучены способы создания запросов, представлений и индексов.