

Министерство науки и высшего образования РФ
Федеральное государственное автономное
образовательное учреждение высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий

институт

Программная инженерия

кафедра

ОТЧЕТ О ПРАКТИЧЕСКОЙ РАБОТЕ

Запросы

тема

Преподаватель

подпись, дата

А. Д. Вожжов

инициалы, фамилия

Студент КИ23-17/16, 032320521

номер группы, зачётной книжки

подпись, дата

А. С. Лысаковский

инициалы, фамилия

Красноярск 2025

1 ВВЕДЕНИЕ

1.1 Цель работы

Изучить теоретический материал по теме «Запросы». Выполнить задания.

1.2 Задачи

В рамках данной практической работы необходимо выполнить следующие задачи:

- 1 изучить теоретический материал по предложенной теме;
- 2 выполнить задание;
- 3 предоставить отчёт преподавателю.

1.3 Задание

Задание данной практической работы состоит из следующих частей:

- 4 Выполнить задания из главы 6 из книги на е-курсах.

2 ХОД РАБОТЫ

2.1 Задание 2

На рисунке 1 показан результат выполнения задания.

```
demo=# SELECT passenger_name FROM tickets WHERE passenger_name LIKE '% ____';
```

Рисунок 1 — Запрос

2.2 Задание 4

На рисунках со 2 по 13 показан результат выполнения задания.

```
demo=# SELECT flight_no  
demo=# FROM flights  
demo=# WHERE flight_no ~* '^[a-z]{2}[0-9]+$';
```

Рисунок 2 — Предикаты, часть 1

```
demo=# SELECT flight_no, scheduled_departure  
demo=# FROM flights  
demo=# WHERE actual_departure IS NULL;
```

Рисунок 3 — Предикаты, часть 2

```
demo=# SELECT ticket_no, passenger_name  
demo=# FROM tickets  
demo=# WHERE passenger_name SIMILAR TO '% S[a-zA-Z]{4}';
```

Рисунок 4 — Предикаты, часть 3

```
demo=# SELECT ticket_no, book_ref  
demo=# FROM tickets  
demo=# WHERE book_ref IN ('ABC123', 'DEF456', 'GHI789');
```

Рисунок 5 — Предикаты, часть 4

```
demo=# SELECT ticket_no, passenger_name  
demo=# FROM tickets  
demo=# WHERE passenger_name ILIKE 'ivan%';
```

Рисунок 6 — Предикаты, часть 5

```
demo=# SELECT flight_no, scheduled_departure, actual_departure  
demo=# FROM flights  
demo=# WHERE scheduled_departure IS NOT DISTINCT FROM actual_departure;
```

Рисунок 7 — Предикаты, часть 6

```
demo=# SELECT ticket_no, passenger_name  
demo=# FROM tickets  
demo=# WHERE passenger_name LIKE 'IVAN%';
```

Рисунок 8 — Предикаты, часть 7

```
demo=# SELECT flight_no, scheduled_departure, actual_departure
demo=# FROM flights
demo=# WHERE scheduled_departure IS DISTINCT FROM actual_departure;
```

Рисунок 9 — Предикаты, часть 8

```
demo=# SELECT aircraft_code, model
demo=# FROM aircrafts
demo=# WHERE aircraft_code NOT IN (
demo(#   SELECT aircraft_code
demo(#   FROM flights
demo(#   WHERE status = 'Cancelled'
demo(# );
```

Рисунок 10 — Предикаты, часть 9

```
demo=# SELECT flight_no, actual_departure
demo=# FROM flights
demo=# WHERE actual_departure IS NOT NULL;
```

Рисунок 11 — Предикаты, часть 10

```
demo=# SELECT flight_no, departure_airport
demo=# FROM flights
demo=# WHERE departure_airport IN (
demo(#   SELECT airport_code
demo(#   FROM airports
demo(#   WHERE city = 'Москва'
demo(# );
```

Рисунок 12 — Предикаты, часть 11

```
demo=# SELECT flight_no, scheduled_departure
demo=# FROM flights
demo=# WHERE flight_no ~ '^[A-Z]{2}[0-9]+$';
```

Рисунок 13 — Предикаты, часть 12

2.3 Задание 6

На рисунке 14 показан результат выполнения задания.

```
demo=# SELECT
demo=# r.flight_no,
demo=# r.departure_airport,
demo=# r.arrival_airport,
demo=# a.model
demo=# FROM routes r
demo=# JOIN aircrafts a ON r.aircraft_code = a.aircraft_code
demo=# WHERE a.model LIKE 'Boeing%';
```

flight_no	departure_airport	arrival_airport	model
PG0013	AER	SVO	Boeing 777-300
PG0073	KRR	NOZ	Boeing 737-300
PG0091	RGK	STW	Boeing 737-300
PG0092	STW	RGK	Boeing 737-300

Рисунок 14 — Запрос

2.4 Задание 8

На рисунке 15 показан результат выполнения задания.

```

demo=# SELECT
demo=#     a.aircraft_code AS aircraft_code_a,
demo=#     a.model,
demo=#     f.flight_no,
demo=#     f.scheduled_departure
demo=# FROM
demo=#     aircrafts a
demo=# FULL OUTER JOIN
demo=#     flights f ON a.aircraft_code = f.aircraft_code;

```

aircraft_code_a	model	flight_no	scheduled_departure
321	Airbus A321-200	PG0405	2016-09-13 12:35:00+07
321	Airbus A321-200	PG0404	2016-10-03 22:05:00+07
321	Airbus A321-200	PG0405	2016-10-03 12:35:00+07
321	Airbus A321-200	PG0402	2016-11-07 15:25:00+07

Рисунок 15 — Запрос

2.5 Задание 10

На рисунке 16 показан результат выполнения задания.

```

demo=# SELECT
demo=#     departure_city,
demo=#     COUNT(DISTINCT arrival_airport) AS destination_count
demo=# FROM
demo=#     routes
demo=# GROUP BY
demo=#     departure_city
demo=# ORDER BY
demo=#     destination_count DESC;

```

departure_city	destination_count
Москва	81
Санкт-Петербург	24
Новосибирск	15

Рисунок 16 — Прогресс

2.6 Задание 12

На рисунке 17, показан результат выполнения задания.

Задание 1. Функция `unnest` раскрывает массив в последовательность строчек.

Задание 2. Показано на рисунке 17.

```

demo=# SELECT
demo-#     dw.name_of_day,
demo-#     COUNT(*) AS num_flights
demo-# FROM
demo-#     (SELECT unnest(days_of_week) AS num_of_day
demo-#       FROM routes
demo-#       WHERE departure_city = 'Москва') AS r
demo-# JOIN
demo-#     unnest('{ "Пн.", "Вт.", "Ср.", "Чт.", "Пт.", "Сб.", "Вс." } '::text[])
demo-#     WITH ORDINALITY AS dw(name_of_day, ordinality)
demo-# ON
demo-#     r.num_of_day = dw.ordinality
demo-# GROUP BY
demo-#     dw.ordinality, dw.name_of_day
demo-# ORDER BY
demo-#     dw.ordinality;
name_of_day | num_flights
-----+-----
Пн.         |         131
Вт.         |         134
Ср.         |         127
Чт.         |         135
Пт.         |         124
Сб.         |         133
Вс.         |         124
(7 строк)

```

Рисунок 17 — Задание 2

2.7 Задание 14

На рисунках 18 показан результат выполнения задания.

```

demo=# SELECT
demo-#     split_part(passenger_name, ' ', 2) AS lastname,
demo-#     count(*) AS occurrences
demo-# FROM bookings.tickets
demo-# GROUP BY 1
demo-# ORDER BY 2 DESC;
lastname | occurrences
-----+-----
IVANOV   |         7539
KUZNECOV |         6906
IVANOVA  |         6501

```

Рисунок 18 — Запрос

2.8 Задание 16

На рисунках 19 показан результат выполнения задания.

Предложение `FILTER` в «PostgreSQL» используется с агрегатными функциями (например, `COUNT`, `SUM`, `AVG`) для указания условия, ограничивающего строки, которые учитываются при вычислении агрегата. Это альтернатива подзапросам или условным выражениям внутри агрегатной функции (например, с `CASE`).

```

demo=# SELECT
demo-#     a.city AS departure_city,
demo-#     COUNT(*) AS total_flights,
demo-#     COUNT(*) FILTER (WHERE f.status = 'Departed') AS departed_flights,
demo-#     COUNT(*) FILTER (WHERE f.status = 'Cancelled') AS cancelled_flights
demo-# FROM
demo-#     routes r
demo-# JOIN
demo-#     flights f ON r.flight_no = f.flight_no
demo-# JOIN
demo-#     airports a ON r.departure_airport = a.airport_code
demo-# WHERE
demo-#     a.city = 'Москва'
demo-# GROUP BY
demo-#     a.city;

```

departure_city	total_flights	departed_flights	cancelled_flights
Москва	7917	15	77

(1 строка)

Рисунок 19 — Применение FILTER

2.9 Задание 18

На рисунке 21 показан результат выполнения задания.

```

demo=# SELECT
demo-#     a.model,
demo-#     COUNT(r.flight_no) AS route_count,
demo-#     ROUND(
demo-#         COUNT(r.flight_no)::numeric /
demo-#         SUM(COUNT(r.flight_no)) OVER () * 100,
demo-#         2
demo-#     ) AS route_percentage
demo-# FROM
demo-#     aircrafts a
demo-# LEFT JOIN
demo-#     routes r ON a.aircraft_code = r.aircraft_code
demo-# GROUP BY
demo-#     a.model
demo-# ORDER BY
demo-#     route_count DESC;

```

model	route_count	route_percentage
Bombardier CRJ-200	232	32.68
Cessna 208 Caravan	170	23.94
Sukhoi SuperJet-100	158	22.25
Airbus A319-100	46	6.48
Boeing 737-300	36	5.07
Airbus A321-200	32	4.51
Boeing 767-300	26	3.66
Boeing 777-300	10	1.41
Airbus A320-200	0	0.00

(9 строк)

Рисунок 20 — Подсчёт процентов

2.10 Задание 20

На рисунках 21 показан результат выполнения задания. COUNT без «b» будет подсчитывать NULL содержащие строки. COUNT с «b» этого делать не будет.

```
demo=# WITH RECURSIVE ranges ( min_sum, max_sum ) AS ( VALU
ES( 0, 100000 ) UNION ALL SELECT min_sum + 100000, max_sum
+ 100000 FROM ranges WHERE max_sum < ( SELECT max( total_am
ount ) FROM bookings ) ) SELECT r.min_sum, r.max_sum, count
( b.* ) FROM bookings b RIGHT OUTER JOIN ranges r ON b.tota
l_amount >= r.min_sum AND b.total_amount < r.max_sum GROUP
BY r.min_sum, r.max_sum ORDER BY r.min_sum;
 min_sum | max_sum | count
-----+-----+-----
         0 | 100000 | 198314
100000   | 200000 | 46943
200000   | 300000 | 11916
300000   | 400000 | 3260
400000   | 500000 | 1357
500000   | 600000 | 681
600000   | 700000 | 222
700000   | 800000 | 55
800000   | 900000 | 24
900000   | 1000000 | 11
1000000  | 1100000 | 4
1100000  | 1200000 | 0
1200000  | 1300000 | 1
(13 строк)
```

Рисунок 21 — Запрос

2.11 Задание 22

На рисунке 22 представлен прогресс выполнения задания.
«COUNT(*)» в запросе нужен, т.к. удаляет дубликаты городов.

```
demo=# SELECT
demo-#     aa.city,
demo-#     aa.airport_code,
demo-#     aa.airport_name
demo-# FROM
demo-#     (SELECT city, count(*)
demo-#     FROM airports
demo-#     GROUP BY city
demo-#     HAVING count(*) > 1) AS a
demo-# JOIN
demo-#     airports AS aa
demo-#     ON a.city = aa.city
demo-# ORDER BY
demo-#     aa.city, aa.airport_name;
  city | airport_code | airport_name
-----+-----+-----
Москва | VKO          | Внуково
Москва | DME          | Домодедово
Москва | SVO          | Шереметьево
Ульяновск | ULV        | Баратаевка
Ульяновск | ULY          | Ульяновск-Восточный
(5 строк)
```

Рисунок 22 — Запрос

2.12 Задание 24

На рисунках 23, 24 показан результат выполнения задания.


```
demo=# SELECT * FROM airports WHERE timezone IN ( 'Asia/Novokuznetsk', 'Asia/Krasnoyarsk' );
SELECT * FROM airports WHERE timezone = ANY ( VALUES ( 'Asia/Novokuznetsk' ), ( 'Asia/Krasnoyarsk' ) );
```

airport_code	airport_name	city	longitude	latitude	timezone
NOZ	Спиченково	Новокузнецк	86.8772	53.8114	Asia/Novokuznetsk
RGK	Горно-Алтайск	Горно-Алтайск	85.833333	51.966667	Asia/Krasnoyarsk
KJA	Емельяново	Красноярск	92.475	56.18	Asia/Krasnoyarsk
SWT	Стрежевой	Стрежевой	77.65	60.716667	Asia/Krasnoyarsk
KEJ	Кемерово	Кемерово	86.107208	55.270094	Asia/Novokuznetsk
ABA	Абакан	Абакан	91.385	53.74	Asia/Krasnoyarsk
BAX	Барнаул	Барнаул	83.538533	53.363775	Asia/Krasnoyarsk
KYZ	Кызыл	Кызыл	94.4006	51.6694	Asia/Krasnoyarsk
TOF	Богашёво	Томск	85.208333	56.380278	Asia/Krasnoyarsk
NSK	Норильск	Норильск	87.332183	69.311053	Asia/Krasnoyarsk

(10 строк)

Рисунок 23 — Запрос

```
demo=# SELECT departure_city, count( * ) FROM routes GROUP BY departure_city HAVING departure_city IN ( SELECT city FROM airports WHERE longitude > 150 ) ORDER BY count DESC;
```

departure_city	count
Анадырь	4
Магадан	3
Петропавловск-Камчатский	1

(3 строки)

Рисунок 24 — Модифицированный запрос

2.13 Задание 26

На рисунке 25 показан результат выполнения задания.

```

demo=# WITH passenger_seats AS (
demo=#     SELECT
demo=#         t.passenger_name,
demo=#         b.seat_no,
demo=#         t.contact_data->'email' AS email,
demo=#         tf.fare_conditions
demo=#     FROM
demo=#         ticket_flights tf
demo=#     JOIN
demo=#         tickets t
demo=#         ON tf.ticket_no = t.ticket_no
demo=#     JOIN
demo=#         boarding_passes b
demo=#         ON tf.ticket_no = b.ticket_no AND tf.flight_id = b.flight_id
demo=#     WHERE
demo=#         tf.flight_id = 27584
demo=# )
demo=# SELECT
demo=#     s.seat_no,
demo=#     p.passenger_name,
demo=#     p.email,
demo=#     p.fare_conditions
demo=# FROM
demo=#     seats s
demo=# LEFT OUTER JOIN
demo=#     passenger_seats p
demo=#     ON s.seat_no = p.seat_no
demo=# WHERE
demo=#     s.aircraft_code = 'SU9'
demo=# ORDER BY
demo=#     left(s.seat_no, length(s.seat_no) - 1)::integer,
demo=#     right(s.seat_no, 1);

```

seat_no	passenger_name	email	fare_conditions
1A	ALEKSANDR ABRAMOV		Business
1C			
1D	DENIS PETROV		Business
1F	LEONID BARANOV	"baranov.1.1967@postgrespro.ru"	Business

Рисунок 25 — Модифицированный запрос

3 ЗАКЛЮЧЕНИЕ

По результатам работы был изучен теоретический материал по теме «Запросы». Все поставленные цели и задачи были выполнены.