

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

Отчет

ПО ЛАБОРАТОРНОЙ РАБОТЕ №2
«ЗАПРОСЫ НА ВЫБОРКУ И МОДИФИКАЦИЮ ДАННЫХ,
ПРЕДСТАВЛЕНИЯ И ИНДЕКСЫ В POSTGRESQL»

Автор: Павлишина Ирина Романовна

Факультет: ИКТ

Группа: К32391

Преподаватель: Говорова М. М.

Дата: 10.05.2023

ИТМО

Санкт-Петербург 2023

Лабораторная №2

Цель работы:

Овладеть практическими навыками создания представлений и запросов на выборку данных к базе данных PostgreSQL, использования подзапросов при модификации данных и индексов.

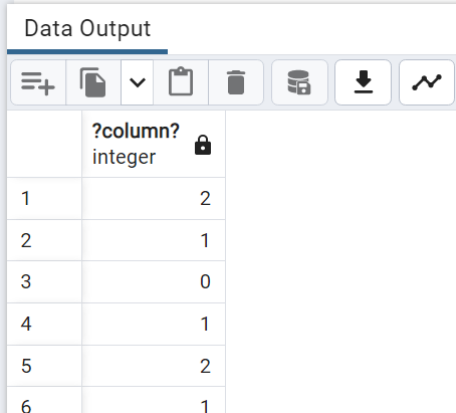
Практическое задание:

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 посмотреть 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.

Запросы

- Определить расчетное время полета по всем маршрутам.

```
SELECT (arrive_date - departure_date) FROM all_flights.schedule;
```



	?column?	integer
1		2
2		1
3		0
4		1
5		2
6		1

- Определить расход топлива по всем маршрутам.

```
SELECT f.id,  
s.departure_date,  
s.arrive_date,  
b.fuel_consumption * s.distance as fuel_consumed
```

```

FROM all_flights.flight f
JOIN all_flights.schedule s ON f.id_schedule = s.id
JOIN all_flights.plane p ON f.id_plane = p.id
JOIN all_flights.brand b ON p.id_brand = b.id;

```

Data Output

	id integer	departure_date date	arrive_date date	fuel_consumed integer
1	1	2024-04-12	2024-04-15	271244
2	2	2024-04-13	2024-04-15	295450
3	3	2024-04-13	2024-04-13	206492
4	4	2024-04-13	2024-04-15	107502
5	5	2024-04-12	2024-04-13	165452
6	6	2024-04-13	2024-04-15	367612
7	7	2024-04-13	2024-04-15	345914

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

```
SELECT COUNT(*) as free_places
```

```
FROM all_flights.flight f
```

```
JOIN all_flights.place p ON f.id = p.id_flight
```

```
WHERE f.flight_num = 'номер_рейса' AND f.date_departure =
CURRENT_DATE - INTERVAL '1 day' AND p.emp_status = true;
```

Data Output

	free_places bigint
1	0

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

```
SELECT SUM(c.cost) as losses
```

```
FROM all_flights.ticket t
```

```
JOIN all_flights.cost c ON t.id_cost = c.id
```

```
JOIN all_flights.flight f ON t.id_flight = f.id
```

WHERE t.payment_status = false AND t.done_status = false AND
f.date_departure = CURRENT_DATE - INTERVAL '1 day';

	losses money
1	[null]

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

```
SELECT b.name, COUNT(*) as flights_count
FROM all_flights.flight f
JOIN all_flights.plane p ON f.id_plane = p.id
JOIN all_flights.brand b ON p.id_brand = b.id
JOIN all_flights.schedule s ON f.id_schedule = s.id
JOIN all_flights.airport a ON s.id_airport = a.id
WHERE a.name = 'аэропорт_назначения'
GROUP BY b.name
ORDER BY flights_count DESC
LIMIT 1;
```

Data Output

	name	flights_count
	character varying (30)	bigint
1	brand_7875	3

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

```
SELECT p.id, p.plane_num, p.creation_date, p.mileage, p.last_repair_date,
p.id_brand, p.id_airline
FROM all_flights.plane p
JOIN all_flights.brand b ON p.id_brand = b.id
WHERE p.creation_date <= (current_date - INTERVAL '1 day')
AND p.creation_date <= (SELECT creation_date FROM all_flights.plane WHERE
id_brand = b.id);
```

Data Output							
	id [PK] integer	plane_num character varying (20)	creation_date date	mileage integer	last_repair_date date	id_brand integer	id_airline integer

- Определить тип самолетов, летающих во все аэропорты назначения.

```
SELECT DISTINCT p.id_brand, b.name
FROM all_flights.plane p
JOIN all_flights.flight f ON p.id = f.id_plane
JOIN all_flights.schedule s ON f.id_schedule = s.id
JOIN all_flights.airport a ON s.id_airport = a.id
JOIN all_flights.brand b ON p.id_brand = b.id
WHERE NOT EXISTS (
    SELECT a2.id
    FROM all_flights.airport a2
    WHERE NOT EXISTS (
        SELECT s2.id
        FROM all_flights.schedule s2
        WHERE s2.id_airport = a2.id AND s2.id = s.id
    )
);
```

Data Output	
id_brand integer	name character varying (30)

Создание представлений

- Для пассажиров авиакомпании о рейсах в Москву на ближайшую неделю;

```

CREATE VIEW all_flights.passengers_to_moscow AS
SELECT p.name, p.passport, f.date_departure, f.date_arrive
FROM all_flights.flight f
JOIN all_flights.ticket t ON t.id_flight = f.id
JOIN all_flights.passanger p ON p.id = t.id_passanger
JOIN all_flights.schedule s ON s.id = f.id_shedule
JOIN all_flights.airport a ON a.id = s.id_flight
WHERE a.city = 'Moscow'
AND f.date_departure BETWEEN CURRENT_DATE AND CURRENT_DATE +
INTERVAL '1 week';

```

- Количество самолетов каждого типа, летавшими за последний месяц.

```

CREATE VIEW all_flights.aircraft_types_count AS
SELECT b.name, COUNT(DISTINCT p.id) AS aircraft_count
FROM all_flights.plane p
JOIN all_flights.brand b ON b.id = p.id_brand
JOIN all_flights.flight f ON f.id_plane = p.id
WHERE f.date_departure BETWEEN CURRENT_DATE - INTERVAL '1 month'
AND CURRENT_DATE
GROUP BY b.name;

```

Вставка значений

- Полет по последнему добавленному расписанию

```

INSERT INTO all_flights.flight ( id_schedule, date_departure, date_arrive,
flight_num, id_plane)
VALUES (
    (SELECT max(id) FROM all_flights.schedule),
    (SELECT (departure_date) FROM all_flights.schedule WHERE id =
(SELECT max(id) FROM all_flights.schedule)),

```

```
(SELECT (arrive_date) FROM all_flights.schedule WHERE (id = (SELECT
max(id) FROM all_flights.schedule))),

'num_test', 10);
```

	id [PK] integer	date_departure date	date_arrive date	flight_num character varying (20)	id_plane integer	id_schedule integer
1	387482	2024-04-11	2024-04-15	num_test	10	400
2	387481	2024-05-11	2024-05-14	400+1000	1000	400
3	387480	2024-05-13	2024-05-15	399+1000	1000	399

Изменение элемента

```
UPDATE all_flights.ticket
SET done_status=true
WHERE id IN (
SELECT t.id FROM all_flights.ticket t
JOIN all_flights.flight fl ON t.id_flight = fl.id
JOIN all_flights.plane pl ON fl.id_plane = pl.id
WHERE pl.plane_num = 'plane_num_10'
);
```

Data Output			
	id integer	done_status boolean	plane_num character varying (20)
1	358	false	plane_num_10
2	363	true	plane_num_10
3	334	false	plane_num_10
4	727	false	plane_num_10
5	410	true	plane_num_10

Data Output			
	id integer	done_status boolean	plane_num character varying (20)
1	358	true	plane_num_10
2	363	true	plane_num_10
3	334	true	plane_num_10
4	727	true	plane_num_10
5	410	true	plane_num_10

Удаление элемента

```
DELETE FROM all_flights.flight WHERE id = (SELECT max(id) FROM all_flights.flight);
```

Data Output						
	id [PK] integer	date_departure date	date_arrive date	flight_num character varying (20)	id_plane integer	id_schedule integer
1	387481	2024-05-11	2024-05-14	400+1000	1000	400
2	387480	2024-05-13	2024-05-15	399+1000	1000	399
3	387479	2024-05-12	2024-05-14	398+1000	1000	398
4	387478	2024-05-13	2024-05-14	397+1000	1000	397
5	387477	2024-05-12	2024-05-13	396+1000	1000	396

Индексы

EXPLAIN

```
SELECT fl.id, fl.id_plane FROM all_flights.flight fl
JOIN all_flights.plane pl ON pl.id = fl.id_plane
```

Data Output	
	QUERY PLAN
	text
1	Hash Join (cost=31.50..7777.75 rows=387481 width=11)
2	Hash Cond: (fl.id_plane = pl.id)
3	-> Seq Scan on flight fl (cost=0.00..6724.81 rows=387481 width=1...
4	-> Hash (cost=19.00..19.00 rows=1000 width=4)
5	-> Seq Scan on plane pl (cost=0.00..19.00 rows=1000 width=4)

Создание индекса

```
CREATE INDEX flight_plane_idx ON all_flights.flight (id, id_plane);
```

CREATE INDEX
Query returned successfully in 239 msec.

Выводы

В процессе выполнения лабораторной работы были созданы генераторы для заполнения базы данных, составлены сложные запросы, запросы INSERT, UPDATE и DELETE, а также изучено графическое представление запросов. Кроме того, были созданы индексы, что привело к уменьшению количества этапов при выполнении запросов и ускорению их выполнения.