

ЗАПРОСЫ НА ВЫБОРКУ И МОДИФИКАЦИЮ ДАННЫХ. ПРЕДСТАВЛЕНИЯ. РАБОТА С ИНДЕКСАМИ

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.

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.

Выполнил: Мищенко Максим K3239

-- 1.1 код, название и город всех аэропортов со статусом 'active'

```
SELECT airport_code,  
       name    AS airport_name,  
       city  
FROM public.airports  
WHERE status = 'active';
```

-- 1.2 идентификатор рейса, дату вылета и расстояние для рейсов компании 'Airline 1', где distance > 1000

```
SELECT f.flight_id,  
       f.departure_datetime_real,  
       f.distance  
FROM public.flights AS f  
JOIN public.planes  AS p ON f.plane_id = p.plane_id  
JOIN public.flight_company AS c ON p.company_id = c.company_id  
WHERE c.name = 'Airline 1'  
      AND f.distance > 1000;
```

-- 1.3 номера и типы мест, которые еще свободны (status='available') на рейсе 1

```
SELECT s.seat_number,  
       s.seat_type  
FROM public.seats AS s  
JOIN public.tickets AS t ON s.seat_id = t.seat_id  
WHERE t.flight_id = 1  
      AND s.status = 'available';
```

2.

2.1

CREATE OR REPLACE VIEW active_airports AS

```
SELECT airport_code,  
       name    AS airport_name,  
       city,  
       country
```

```
FROM public.airports  
WHERE status = 'active';
```

```
select * from active_airports
```

2.2

```
CREATE OR REPLACE VIEW long_distance_flights AS
SELECT f.flight_id,
       c.name   AS company_name,
       f.distance,
       f.departure_datetime_real,
       f.arrival_datetime_real
FROM public.flights AS f
JOIN public.planes   AS p ON f.plane_id = p.plane_id
JOIN public.flight_company AS c ON p.company_id = c.company_id
WHERE f.distance > 5000;
```

```
select * from long_distance_flights
```

2.3

```
CREATE OR REPLACE VIEW passenger_tickets AS
SELECT t.ticket_id,
       p.full_name AS passenger_name,
       f.flight_id,
       s.seat_number,
       t.status     AS ticket_status
FROM public.tickets AS t
JOIN public.passengers AS p ON t.passenger_id = p.passenger_id
JOIN public.seats     AS s ON t.seat_id = s.seat_id
JOIN public.flights   AS f ON t.flight_id = f.flight_id;
```

```
select * from passenger_tickets
```

3

3.1

INSERT нового рейса по самому дальнему маршруту

```
WITH
max_plane AS (
  SELECT plane_id
  FROM public.planes
```

```

ORDER BY flight_hours DESC
LIMIT 1
),
latest_route AS (
  SELECT route_id
  FROM public.schedule
  ORDER BY planned_time_departure DESC
  LIMIT 1
),
max_dist AS (
  SELECT MAX(distance)::int AS distance
  FROM public.flights
)
INSERT INTO public.flights (
  plane_id,
  route_id,
  status,
  distance,
  departure_datetime_real,
  arrival_datetime_real
)
SELECT
  mp.plane_id,
  lr.route_id,
  'scheduled',
  md.distance,
  now() + interval '1 day',
  now() + interval '1 day' + (md.distance / 500 || ' hours')::interval
FROM max_plane mp, latest_route lr, max_dist md;

```

3.2

UPDATE: пометить все места как 'reserved' для рейсов компании 'Airline 2'

```

UPDATE public.seats AS s
SET status = 'reserved'
WHERE seat_id IN (
  SELECT t.seat_id
  FROM public.tickets AS t
  JOIN public.flights AS f ON t.flight_id = f.flight_id
  JOIN public.planes AS p ON f.plane_id = p.plane_id
  JOIN public.flight_company AS c ON p.company_id = c.company_id
  WHERE c.name = 'Airline 2'
);

```

3.3

DELETE устаревших отмененных билетов

```
DELETE FROM public.tickets
WHERE status = 'cancelled'
AND created_at < now() - INTERVAL '30 days';
```

4

4.1

```
EXPLAIN ANALYZE
SELECT flight_id, distance
FROM public.flights
WHERE distance > 1000;
```

"Execution Time: 0.020 ms"

```
CREATE INDEX idx_flights_distance ON public.flights(distance);
```

```
EXPLAIN ANALYZE
SELECT flight_id, distance
FROM public.flights
WHERE distance > 1000;
```

"Execution Time: 0.010 ms"

4.2

```
EXPLAIN ANALYZE
SELECT *
FROM public.tickets
WHERE flight_id = 1
AND passenger_id = 10;
```

"Execution Time: 0.023 ms"

```
CREATE INDEX idx_tickets_flight_passenger ON public.tickets(flight_id, passenger_id);
```

```
EXPLAIN ANALYZE
SELECT *
```

```
FROM public.tickets  
WHERE flight_id = 1  
AND passenger_id = 10;
```

"Execution Time: 0.012 ms"

Вывод

1. Представления упрощают повторное использование сложных запросов, скрывая детали соединений.
2. Подзапросы в модификациях позволяют динамически извлекать значения (например, по максимальному пробегу).
3. Создание индексов существенно ускоряет выборки по полям, участвующим в условиях WHERE, особенно для больших таблиц.