

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

по дисциплине «Проектирование и реализация баз данных»

Автор: Казанков И

Факультет: ИКТ

Группа: К32402

Преподаватель: Говорова М. М.

Санкт-Петербург 2023

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

Программное обеспечение: СУБД PostgreSQL 1X, pgAdmin 4.

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.

Выполнение:

Заполнение бд :

```
INSERT INTO regular_schedule (arrival_time, departure_time, train_arrival_point, train_departure_point)
VALUES
('08:00', '08:30', 'Station A', 'Москва'),
('10:15', '10:45', 'Station C', 'Москва'),
('12:30', '13:00', 'Station E', 'Москва');
```

```
INSERT INTO train (train_name, train_number, fact_time_sent, fact_time_transformation, arrival_date,
departure_date, type_train)
VALUES
('Train A', 1, '07:45', '08:15', '2023-06-29', '2023-06-29', 'Express'),
('Train B', 2, '10:00', '10:30', '2023-06-28', '2023-06-29', 'Local'),
('Train C', 3, '12:15', '12:45', '2023-06-29', '2023-06-29', 'Express');
```

```
INSERT INTO stopover (train_number, type_locality, parking_time, type_stop)
VALUES
(1, 'Irkutsk', '10 minutes', 'Departure'),
(2, 'Krasnoyarsk', '5 minutes', 'Arrival'),
(3, 'Omsk', '15 minutes', 'Departure');
```

```
INSERT INTO station (name_station)
VALUES
('Station A'),
('Station B'),
('Station C'),
('Москва');
```

```
INSERT INTO vagon (train_id, vagon_type, vagon_number_in_train, counter_place)
VALUES
(1, 'CB', 1, 36),
(1, 'CB', 2, 54),
(2, 'купейный', 1, 18),
```

```
(2, 'купейный', 2, 54),  
(3, 'CB', 1, 54);
```

```
INSERT INTO place (wagon_id, number_of_seats, occupancy)  
VALUES  
(1, 50, 'Occupied'),  
(1, 50, 'free'),  
(2, 40, 'Occupied'),  
(2, 40, 'Occupied'),  
(3, 30, 'free'),  
(3, 20, 'free');
```

```
INSERT INTO passenger (passport, full_name)  
VALUES  
(ABC123, 'Ivan'),  
(DEF456, 'Ilya'),  
(GHI789, 'Sergey');
```

```
INSERT INTO cashbox (locality, address_cashbox)  
VALUES  
(City X, '123 Main Street'),  
(City Y, '456 Park Avenue');
```

```
INSERT INTO ticket (place_id, stopover_id, cashbox_id, passenger_id, payment_method, date_of_purchase,  
price_ticket, arrival_point, departure_point)  
VALUES  
(1, 1, 1, 1, 'Cash', '2023-06-28', 10, 'Station A', 'Москва'),  
(2, 2, 2, 2, 'Credit Card', '2023-06-28', 15, 'Station A', 'Москва'),  
(3, 3, 1, 3, 'Cash', '2023-06-27', 20, 'Station A', 'Москва');
```

1: Запросы:

1 - Свободные места на все поезда, отправляющиеся с вокзала в течение следующих суток.

```
1 SELECT t.train_number, v.vagon_number_in_train, p.number_of_seats  
2 FROM train t  
3 JOIN vagon v ON t.train_id = v.train_id  
4 JOIN place p ON v.vagon_id = p.vagon_id  
5 WHERE t.departure_date_time >= CURRENT_TIMESTAMP  
6 AND p.occupancy = 'free';
```

Data Output

Сообщения

Notifications

	train_number integer	vagon_number_in_train integer	number_of_seats integer
1	1	1	50
2	2	1	20
3	2	1	30

2 - Список пассажиров, отправившихся в Москву всеми рейсами за прошедшие сутки.

```
1 SELECT p.full_name, p.passport
2 FROM passenger p
3 JOIN ticket t ON p.passenger_id = t.passenger_id
4 JOIN arrival_point ap ON t.arrival_point = ap.arrival_point_name
5 JOIN regular_schedule rs ON ap.train_number = rs.train_number
6 WHERE t.departure_point = 'Москва'
7       AND t.date_of_purchase >= CURRENT_DATE - INTERVAL '1 day'
8       AND t.date_of_purchase < CURRENT_DATE
```

Data Output Сообщения Notifications

	full_name character varying (40) 🔒	passport character varying (10) 🔒
1	Ivan	ABC123
2	Ilya	DEF456
3	Sergey	GHI789

3 - Номера поездов, на которые проданы все билеты на следующие сутки.

```
1 WITH total_places_per_train AS (  
2   SELECT train_id, SUM(counter_place) AS total_places  
3   FROM vagon  
4   GROUP BY train_id  
5 ),  
6 occupied_counts_per_train AS (  
7   SELECT v.train_id, COUNT(*) AS occupied_count  
8   FROM vagon v  
9   JOIN place p ON v.vagon_id = p.vagon_id  
10  JOIN ticket t ON p.place_id = t.place_id  
11  WHERE p.occupancy = 'Occupied' AND t.date_of_purchase = CURRENT_DATE + INTERVAL '1 day'  
12  GROUP BY v.train_id  
13 )  
14 SELECT tp.train_id, tp.total_places, oc.occupied_count  
15 FROM total_places_per_train tp  
16 JOIN occupied_counts_per_train oc ON tp.train_id = oc.train_id  
17 WHERE tp.total_places = oc.occupied_count;
```

Data Output Сообщения Notifications



train_id	total_places	occupied_count
integer	bigint	bigint

4 - Свободные места в купейные вагоны всех рейсов до Москвы на текущие сутки.

```
1 SELECT t.train_id, COUNT(*) AS free_seats  
2 FROM train t  
3 JOIN vagon v ON t.train_id = v.train_id  
4 JOIN place p ON v.vagon_id = p.vagon_id  
5 JOIN stopover s ON t.train_id = s.train_number  
6 JOIN regular_schedule rs ON s.train_number = rs.train_number  
7 WHERE t.arrival_date = CURRENT_DATE  
8   AND rs.train_departure_point = 'Москва'  
9   AND v.vagon_type = 'купейный'  
10  AND p.occupancy = 'free'  
11 GROUP BY t.train_id;
```

Data Output Сообщения Notifications



	train_id	free_seats
	[PK] integer	bigint
1	2	2

5 - Выручка от продажи билетов на все поезда за прошедшие сутки.

1	SELECT t.train_id, SUM(ti.price_ticket) AS total_revenue
2	FROM train t
3	JOIN vagon v ON t.train_id = v.train_id
4	JOIN place p ON v.vagon_id = p.vagon_id
5	JOIN ticket ti ON p.place_id = ti.place_id
6	WHERE ti.date_of_purchase >= CURRENT_DATE - INTERVAL '1 day'
7	AND ti.date_of_purchase < CURRENT_DATE
8	GROUP BY t.train_id;

Data Output	Сообщения	Notifications
<div> <div>≡+</div> <div>📄</div> <div>▼</div> <div>📋</div> <div>▼</div> <div>🗑️</div> <div>🗑️</div> <div>📥</div> <div>📥</div> <div>📈</div> </div>		
	train_id [PK] integer	total_revenue bigint
1	1	20

6 - Общее количество билетов, проданных по всем направлениям в вагоны типа “СВ”.

1	SELECT COUNT(*) AS total_tickets
2	FROM ticket ti
3	JOIN place p ON ti.place_id = p.place_id
4	JOIN vagon v ON p.vagon_id = v.vagon_id
5	WHERE v.vagon_type = 'CB';

Data Output	Сообщения	Notifications
<div> <div>≡+</div> <div>📄</div> <div>▼</div> <div>📋</div> <div>▼</div> <div>🗑️</div> <div>🗑️</div> <div>📥</div> <div>📥</div> <div>📈</div> </div>		
	total_tickets bigint	
1	3	

7 - Номера и названия поездов, все вагоны которых были заполнены менее чем наполовину за прошедшие сутки.

Для всего поезда:

```

1 WITH total_occupied_per_train AS (
2     SELECT v.train_id, COUNT(*) AS occupied_count
3     FROM vagon v
4     JOIN place p ON v.vagon_id = p.vagon_id
5     WHERE p.occupancy = 'Occupied'
6     GROUP BY v.train_id
7 ),
8 total_places_per_train AS (
9     SELECT train_id, SUM(counter_place) AS total_places
10    FROM vagon
11    GROUP BY train_id
12 ),
13 under_half_filled_trains AS (
14     SELECT t.train_id
15     FROM total_occupied_per_train t
16     JOIN total_places_per_train p ON t.train_id = p.train_id
17     WHERE t.occupied_count < p.total_places / 2
18 )
19 SELECT tr.train_number, tr.train_name
20 FROM train tr
21 JOIN under_half_filled_trains u ON tr.train_id = u.train_id
22 WHERE tr.departure_date_time >= CURRENT_DATE - INTERVAL '1 day';

```

Data Output Сообщения Notifications



	train_number integer	train_name character varying (20)
1	1	Train A

Для каждого вагона

```
1 WITH occupied_places AS (  
2     SELECT v.vagon_id, COUNT(*) AS occupied_count  
3     FROM vagon v  
4     JOIN place p ON v.vagon_id = p.vagon_id  
5     WHERE p.occupancy = 'Occupied'  
6     GROUP BY v.vagon_id  
7 ),  
8 total_places_per_train AS (  
9     SELECT v.train_id, SUM(v.counter_place) AS total_places  
10    FROM vagon v  
11    GROUP BY v.train_id  
12 ),  
13 underfilled_trains AS (  
14     SELECT tp.train_id  
15    FROM total_places_per_train tp  
16    JOIN occupied_places op ON tp.train_id = op.vagon_id  
17    WHERE op.occupied_count < (tp.total_places / 2)  
18 )  
19 SELECT t.train_number, t.train_name  
20 FROM train t  
21 WHERE t.train_id IN (  
22     SELECT train_id  
23     FROM underfilled_trains  
24 );
```


представление:

для пассажиров о наличии свободных мест на заданный рейс:

```
1 CREATE VIEW free_seats AS
2 SELECT p.place_id, v.vagon_type, v.vagon_number_in_train, p.number_of_seats
3 FROM place p
4 INNER JOIN vagon v ON p.vagon_id = v.vagon_id
5 WHERE p.occupancy = 'free';
```

Data Output Сообщения Notifications

CREATE VIEW

Запрос завершён успешно, время выполнения: 418 msec.

количество непроданных билетов на все поезда, формирующиеся за прошедшие сутки (номер поезда, тип вагона, количество).

```
CREATE INDEX idx_train_departure_date ON train (departure_date);
CREATE INDEX idx_vagon_train_id ON vagon (train_id);
CREATE INDEX idx_place_vagon_id_occupancy ON place (vagon_id, occupancy);
EXPLAIN SELECT t.train_id, COUNT(*) AS free_seats
FROM train t
JOIN vagon v ON t.train_id = v.train_id
JOIN place p ON v.vagon_id = p.vagon_id
WHERE t.departure_date = CURRENT_DATE + INTERVAL '1 day'
AND p.occupancy = 'free'
GROUP BY t.train_id;
```

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

INSERT:

```
1 INSERT INTO passenger (passport, full_name)
2 SELECT '1234567890', 'Ilya Kazankov'
3 WHERE NOT EXISTS (SELECT 1 FROM passenger WHERE passport = '1234567890');
```

Data Output Сообщения Notifications

INSERT 0 1

Запрос завершён успешно, время выполнения: 35 msec.

	passenger_id [PK] integer	passport character varying (10)	full_name character varying (40)
1	1	ABC123	Ivan
2	2	DEF456	Ilya
3	3	GHI789	Sergey
4	4	1234567890	Ilya Kazankov

UPDATE:

```
1 UPDATE place
2 SET occupancy = 'Occupied'
3 WHERE vagon_id = (
4     SELECT vagon_id
5     FROM vagon
6     WHERE vagon_type = 'купейный'
7     LIMIT 1
8 );
```

Data Output Сообщения Notifications

UPDATE 2

Запрос завершён успешно, время выполнения: 53 msec.

DELETE:

```
1 DELETE FROM vagon
2 WHERE train_id = (
3     SELECT train_id
4     FROM train
5     WHERE train_number = 123
6 );
```

Data Output Сообщения Notifications

DELETE 0

Запрос завершён успешно, время выполнения: 415 msec.

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:

1 запрос:

```
1 SELECT t.train_id, COUNT(*) AS free_seats
2 FROM train t
3 JOIN vagon v ON t.train_id = v.train_id
4 JOIN place p ON v.vagon_id = p.vagon_id
5 WHERE t.departure_date = CURRENT_DATE + INTERVAL '1 day'
6     AND p.occupancy = 'free'
7 GROUP BY t.train_id;
```

Data Output Сообщения Notifications

	train_id [PK] integer	free_seats bigint
1	1	1
2	2	2

✓ Запрос выполнен успешно. Общее время выполнения: 124 msec. обработано строк: 1. ✕

```
CREATE INDEX idx_train_departure_date ON train (departure_date);
CREATE INDEX idx_vagon_train_id ON vagon (train_id);
CREATE INDEX idx_place_vagon_id_occupancy ON place (vagon_id, occupancy);
```

```
EXPLAIN SELECT t.train_id, COUNT(*) AS free_seats
FROM train t
JOIN vagon v ON t.train_id = v.train_id
JOIN place p ON v.vagon_id = p.vagon_id
WHERE t.departure_date = CURRENT_DATE + INTERVAL '1 day'
AND p.occupancy = 'free'
GROUP BY t.train_id;
```

✓ Запрос выполнен успешно. Общее время выполнения: 55 msec. обработано строк: 13. ✕

2 запрос:

```
1 SELECT pa.full_name
2 FROM passenger pa
3 JOIN ticket ti ON pa.passenger_id = ti.passenger_id
4 JOIN stopover s ON ti.stopover_id = s.stopover_id
5 JOIN regular_schedule rs ON s.train_number = rs.train_number
6 WHERE ti.date_of_purchase >= CURRENT_DATE - INTERVAL '1 day'
7       AND ti.date_of_purchase < CURRENT_DATE
8       AND ti.departure_point = 'Москва';
```

Data Output Сообщения Notifications

	full_name character varying (40) 🔒
1	Sergey

✓ Запрос выполнен успешно. Общее время выполнения: 62 msec. обработано строк: 1. ✕

```
CREATE INDEX idx_passenger_passenger_id ON passenger (passenger_id);
CREATE INDEX idx_ticket_date_of_purchase ON ticket (date_of_purchase);
CREATE INDEX idx_stopover_train_number ON stopover (train_number);
CREATE INDEX idx_regular_schedule_train_departure_point ON regular_schedule (train_departure_point);
```

```
EXPLAIN SELECT pa.full_name
FROM passenger pa
JOIN ticket ti ON pa.passenger_id = ti.passenger_id
JOIN stopover s ON ti.stopover_id = s.stopover_id
JOIN regular_schedule rs ON s.train_number = rs.train_number
WHERE ti.date_of_purchase >= CURRENT_DATE - INTERVAL '1 day'
AND ti.date_of_purchase < CURRENT_DATE
AND rs.train_departure_point = 'Москва';
```

✓ Запрос выполнен успешно. Общее время выполнения: 45 msec. обработано строк: 12. ✕

Выводы:

В процессе работы с pgAdmin я научился делать запросы на выборку и модификацию данных в базе данных PostgreSQL. Я освоил создание представлений, которые помогают упростить работу с данными, а также настройку индексов, что повышает производительность работы с базой данных. Эти знания помогут мне более эффективно работать с PostgreSQL и создавать более сложные запросы и представления для обработки большого объема данных. В целом, работа с pgAdmin позволила мне более полно использовать возможности PostgreSQL и повысить эффективность работы с базой данных.