

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 «**Базы данных**»

Автор: Казанков И
Факультет: ИКТ
Группа: K32402
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 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.

Выполнение:

1: Запросы:

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

```
SELECT t.number_train, COUNT(*) AS free_seats
FROM train t
LEFT JOIN railway_carriage rc ON t.number_train = rc.number_train
LEFT JOIN place p ON rc.number_railway_carriage = p.number_railway_carriage
LEFT JOIN ticket ti ON p.number_place = ti.number_place AND ti.number_train = t.number_train
WHERE ti.number_of_ticket IS NULL
      AND t.date_and_time_sent >= CURRENT_DATE
      AND t.date_and_time_sent < CURRENT_DATE + INTERVAL '1 day'
GROUP BY t.number_train;
```

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

```
SELECT p.full_name
FROM passenger p
JOIN ticket ti ON p.passenger_id = ti.passenger_id
JOIN stop s ON ti.schedule_id = s.schedule_id AND ti.number_train = s.number_train
JOIN station st ON s.number_station = st.number_station
WHERE st.name_station = 'Москва'
      AND ti.date_purchases >= CURRENT_DATE - INTERVAL '1 day'
      AND ti.date_purchases < CURRENT_DATE;
```

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

```
SELECT t.number_train
FROM train t
JOIN ticket ti ON t.number_train = ti.number_train
WHERE ti.date_purchases >= CURRENT_DATE
      AND ti.date_purchases < CURRENT_DATE + INTERVAL '1 day'
GROUP BY t.number_train
HAVING COUNT(*) = (SELECT COUNT(*) FROM place WHERE number_train = t.number_train);
```

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

```
SELECT t.number_train, rc.number_railway_carriage, COUNT(*) AS free_seats
FROM train t
JOIN railway_carriage rc ON t.number_train = rc.number_train
JOIN place p ON rc.number_railway_carriage = p.number_railway_carriage
LEFT JOIN ticket ti ON p.number_place = ti.number_place AND ti.number_train = t.number_train
JOIN stop s ON ti.schedule_id = s.schedule_id AND ti.number_train = s.number_train
JOIN station st ON s.number_station = st.number_station
WHERE st.name_station = 'Москва'
      AND t.date_and_time_sent >= CURRENT_DATE
      AND t.date_and_time_sent < CURRENT_DATE + INTERVAL '1 day'
      AND ti.number_of_ticket IS NULL
      AND p.location_type = 'Kupe'
GROUP BY t.number_train, rc.number_railway_carriage;
```

Выручка от продажи билетов на все поезда за прошедшие сутки.

```
SELECT t.number_train, SUM(ti.price_ticket) AS total_revenue
FROM train t
JOIN ticket ti ON t.number_train = ti.number_train
WHERE ti.date_purchases >= CURRENT_DATE - INTERVAL '1 day'
      AND ti.date_purchases < CURRENT_DATE
GROUP BY t.number_train;
```

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

```
SELECT COUNT(*) AS total_tickets
FROM ticket ti
JOIN place p ON ti.number_place = p.number_place
JOIN railway_carriage rc ON p.number_railway_carriage = rc.number_railway_carriage
WHERE rc.railway_carriage_type = 'CB';
```

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

```
SELECT t.number_train, t.name_train
FROM train t
JOIN railway_carriage rc ON t.number_train = rc.number_train
JOIN place p ON rc.number_railway_carriage = p.number_railway_carriage
LEFT JOIN ticket ti ON p.number_place = ti.number_place AND ti.number_train = t.number_train
WHERE ti.date_purchases >= CURRENT_DATE - INTERVAL '1 day'
      AND ti.date_purchases < CURRENT_DATE
GROUP BY t.number_train, t.name_train
HAVING COUNT(ti.number_of_ticket) < (SELECT COUNT(*) FROM place WHERE number_train = t.number_train) / 2;
```

представление:

Запрос История запросов

```
1 CREATE VIEW available_seats_view AS
2 SELECT t.number_train, t.name_train, rc.railway_carriage_type,
3 FROM train t
4 JOIN railway_carriage rc ON t.number_train = rc.number_train
5 JOIN place p ON rc.number_railway_carriage = p.number_railway_carriage
6 LEFT JOIN ticket ti ON p.number_place = ti.number_place AND ti
7 WHERE ti.number_of_ticket IS NULL
8 GROUP BY t.number_train, t.name_train, rc.railway_carriage_type
```

Data Output Сообщения Notifications

CREATE VIEW

Запрос завершён успешно, время выполнения: 64 msec.

Запрос История запросов

```
1 CREATE VIEW unsold_tickets_view AS
2 SELECT t.number_train, rc.railway_carriage_type, COUNT(*) AS unsold_tickets
3 FROM train t
4 JOIN railway_carriage rc ON t.number_train = rc.number_train
5 JOIN place p ON rc.number_railway_carriage = p.number_railway_carriage
6 LEFT JOIN ticket ti ON p.number_place = ti.number_place AND ti
7 WHERE t.date_and_time_sent >= CURRENT_DATE - INTERVAL '1 day'
8 AND t.date_and_time_sent < CURRENT_DATE
9 AND ti.number_of_ticket IS NULL
10 GROUP BY t.number_train, rc.railway_carriage_type;
```

Data Output Сообщения Notifications

CREATE VIEW

Запрос завершён успешно, время выполнения: 41 msec.

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

Запрос

История запросов

↗

S

```
1 INSERT INTO railway_carriage (number_train, railway_carriage_type, number_of_seats, railway_carriage_number_in
2 VALUES (
3     (SELECT number_train FROM train WHERE name_train = 'Express'),
4     'Купе',
5     (SELECT COUNT(*) FROM place WHERE number_railway_carriage = 1),
6     2
7 );
8
```

Data Output

Сообщения

Notifications

INSERT 0 1

Запрос завершён успешно, время выполнения: 113 мсек.

Запрос

История запросов

↗

```
1 UPDATE stop
2 SET duration_stop = '01:30:00'
3 WHERE schedule_id = (SELECT schedule_id FROM schedule WHERE paragraph_departures = 'Departure 1')
4 AND number_station = (SELECT number_station FROM station WHERE name_station = 'Moscow');
5
```

Data Output

Сообщения

Notifications

UPDATE 0

Запрос завершён успешно, время выполнения: 38 мсек.

```
1 DELETE FROM ticket
2 WHERE number_train IN (SELECT number_train FROM train WHERE train_type = 'Express')
3 AND number_place IN (SELECT number_place FROM place WHERE location_type = 'Kyne');
4
```

Data Output Сообщения Notifications

DELETE 0

Запрос завершён успешно, время выполнения: 38 мсек.

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:

```
EXPLAIN SELECT t.number_train, COUNT(*) AS free_seats
FROM train t
LEFT JOIN railway_carriage rc ON t.number_train = rc.number_train
LEFT JOIN place p ON rc.number_railway_carriage = p.number_railway_carriage
LEFT JOIN ticket ti ON p.number_place = ti.number_place AND ti.number_train =
t.number_train
WHERE ti.number_of_ticket IS NULL
AND t.date_and_time_sent >= CURRENT_DATE
AND t.date_and_time_sent < CURRENT_DATE + INTERVAL '1 day'
GROUP BY t.number_train;
```

✓ Запрос выполнен успешно. Общее время выполнения: 81 мсек.
обработано строк: 0. ✕

```
CREATE INDEX idx_free_seats ON ticket (number_train, number_railway_carriage,
number_place);
```

✓ Запрос выполнен успешно. Общее время выполнения: 52 мсек.
обработано строк: 19. ✕

```
EXPLAIN SELECT COUNT(*) AS total_tickets
FROM ticket ti
JOIN place p ON ti.number_place = p.number_place
JOIN railway_carriage rc ON p.number_railway_carriage = rc.number_railway_carriage
WHERE rc.railway_carriage_type = 'CB';
```

✓ Запрос выполнен успешно. Общее время выполнения: 94 msec.
обработано строк: 1. ✕

```
CREATE INDEX idx_sv_tickets ON railway_carriage (railway_carriage_type);
```

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.