МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«МИРЭА – Российский технологический университет»

РТУ МИРЭА

Институт информационных технологий (ИИТ)
Кафедра цифровой трансформации (ЦТ)

ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ №8

по дисциплине «Разработка баз данных»

Студент группы *ИКБО-50-23. Павлов Н.С..* _____
(подпись)

Преподаватель *Мажей Я. В.*

_____
(подпись)

Москва 2025 г.

СОДЕРЖАНИЕ

1. ПОСТАНОВКА ЗАДАЧИ

Цель: освоение методов работы со сложными и неструктурированными типами данных (JSONB, BYTEA) в PostgreSQL, а также получение практических навыков реализации горизонтального масштабирования базы данных с помощью декларативного секционирования.

Задачи:

Задание №1: хранение паролей (BYTEA)

1. Создать справочную таблицу ролей и таблицу системных пользователей (или модифицировать существующую).
2. В таблице пользователей предусмотреть поля для хранения «сырого» пароля (только для учебной демонстрации) и его хеша (тип BYTEA).
3. Сохранить хеши паролей пользователей, используя функцию хеширования digest.
4. Выполнить запрос, проверяющий соответствие введенного пароля сохраненному хешу. Одна проверка должна быть успешной, другая - нет (допустимо сделать это в одном запросе).

Задание №2: секционирование таблицы логов

1. Создать новую секционированную таблицу для хранения логов событий.
2. Использовать стратегию секционирования по диапазонам (RANGE) по полю даты.
3. Создать две секции для периодов:
    - 2-е полугодие 2024 (с 2024-07-01 по 2025-01-01).
    - 1-е полугодие 2025 (с 2025-01-01 по 2025-07-01).
4. Создать секцию по умолчанию, куда будут попадать все данные, выходящие за рамки указанных диапазонов (будущее время).

Задание №3: генерация данных

Написать скрипт на PL/pgSQL для генерации 20 000 записей в диапазоне дат, покрывающем созданные секции. Сгенерированные данные должны содержать JSON-структуру с различным набором полей для разных типов

событий (например, чтобы логи «продажи» отличались по структуре от логов «входа»).

Вывести сгенерированные данные – показать запросом SELECT, что данные успешно созданы и физически распределились по разным таблицам секциям.

Задание №4: анализ и поиск по JSONB

Написать три запроса:

1. Фильтрация по значению – найти записи, где числовое поле внутри JSON удовлетворяет математическому условию (например, > 800), используя операторы извлечения -> и ->>.
2. Поиск по вхождению – найти записи, содержащие точный фрагмент JSON-структуры (например, {"quantity": 5}), используя оператор @>.
3. Агрегация – посчитать сумму или среднее значение числового поля из JSON-документа.

Задание №5: модификация данных JSONB

Продемонстрировать изменение данных внутри JSON-поля:

1. Массовое добавление нового поля во все записи определенного типа.
2. Точечное изменение значения по ключу в конкретной записи.

2 ВЫПОЛНЕНИЕ РАБОТЫ

2.1 ИСХОДНЫЕ ТАБЛИЦЫ

employee | Введите SQL выражение чтобы отфильтровать результаты

	employee_id	position_id	manager_id	surname	name	fathers_name	phone	registration_address	employment_dat
1	1	7	[NULL]	Волков	Александр	Михайлович	+79167654321	г. Москва, ул. Пушкина, 15	2022-
2	2	3	1	Соколов	Игорь	Васильевич	+79167654322	г. Москва, ул. Гагарина, 20	2022-
3	3	2	2	Кузнецов	Павел	Сергеевич	+79167654323	г. Москва, ул. Лермонтова,	2022-
4	4	1	2	Попов	Михаил	Андреевич	+79167654324	г. Москва, ул. Чехова, 30	2022-
5	5	4	2	Васильев	Денис	Олегович	+79167654325	г. Москва, ул. Толстого, 35	2022-
6	6	6	1	Петров	Артем	Викторович	+79167654326	г. Москва, ул. Достоевского,	2022-
7	7	5	6	Михайлов	Кирилл	[NULL]	+79167654327	г. Москва, ул. Тургенева, 45	2022-
8	8	8	2	Новиков	Евгений	Алексеевич	+79167654328	г. Москва, ул. Гоголя, 50	2022-
9	9	2	2	Фролов	Роман	Дмитриевич	+79167654329	г. Москва, ул. Некрасова, 55	2022-
10	10	2	2	Алексеев	Владимир	Павлович	+79167654330	г. Москва, ул. Островского,	2022-

Рисунок 1 – Таблица employee

position | Введите SQL выражение чтобы отфильтровать результаты

	position_id	title	salary
1	1	Мастер-приемщик	60 000
2	2	Веломеханик	55 000
3	3	Старший механик	70 000
4	4	Диагност	65 000
5	5	Менеджер по закупкам	58 000
6	6	Администратор	45 000
7	7	Главный механик	80 000
8	8	Специалист по электронике	72 000

Рисунок 2 – Таблица position

2.2 ЗАДАНИЕ 1. ХРАНЕНИЕ ПАРОЛЕЙ (BYTEA)

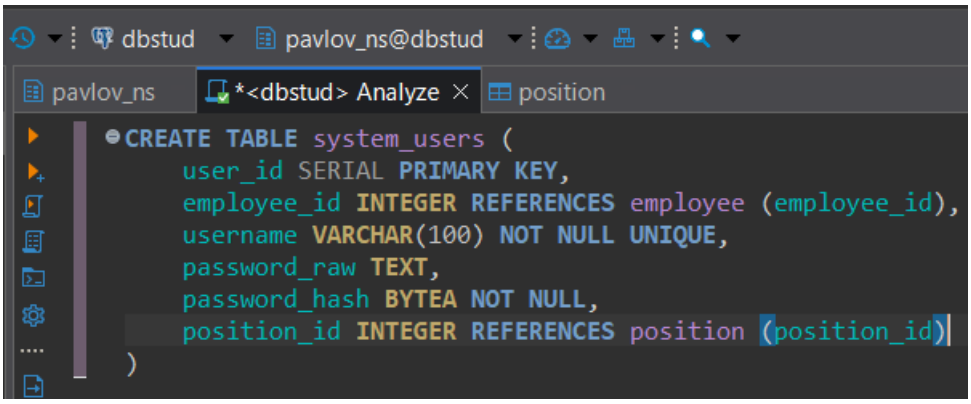

Рисунок 3 – Создание таблицы пользователей

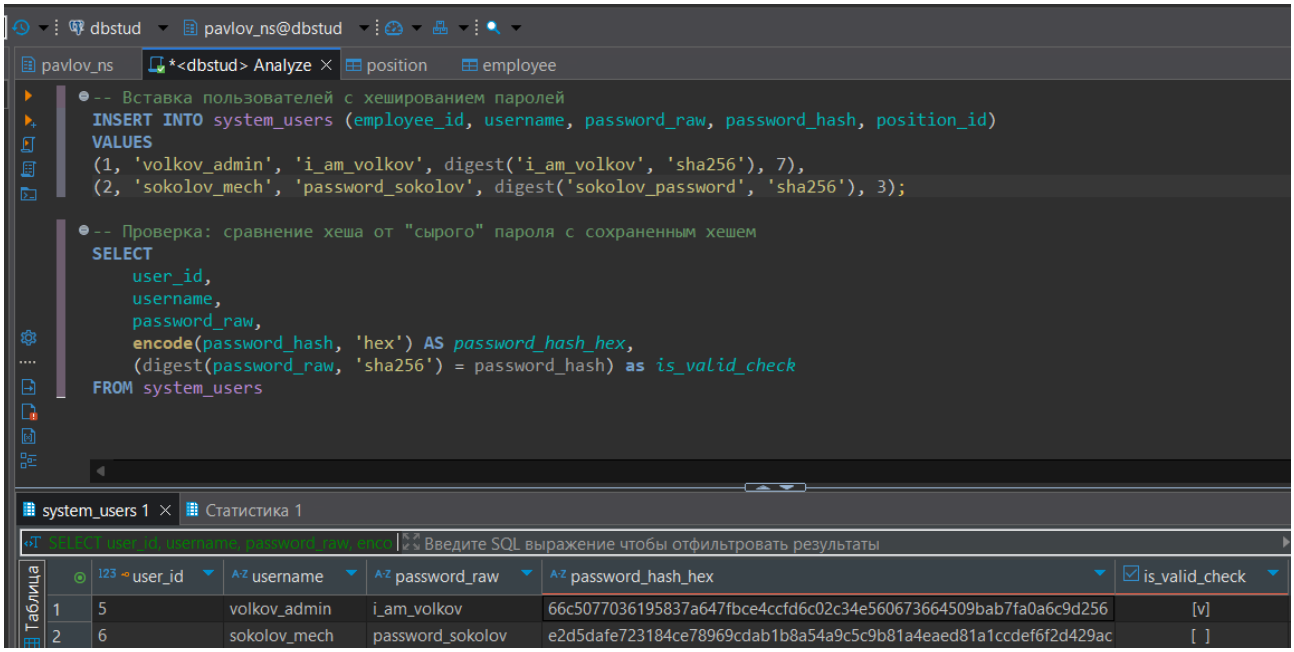

Рисунок 4 – Вставка и проверка данных пользователей

2.3 ЗАДАНИЕ 2. СЕКЦИОНИРОВАНИЕ ТАБЛИЦЫ ЛОГОВ

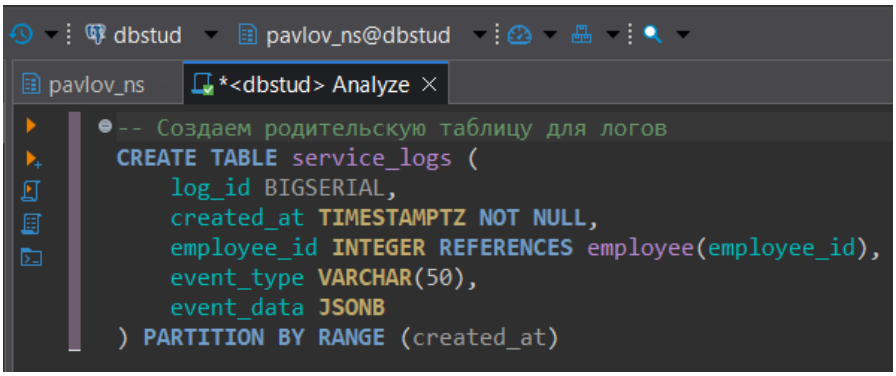

Рисунок 5 – Создание секционированной таблицы

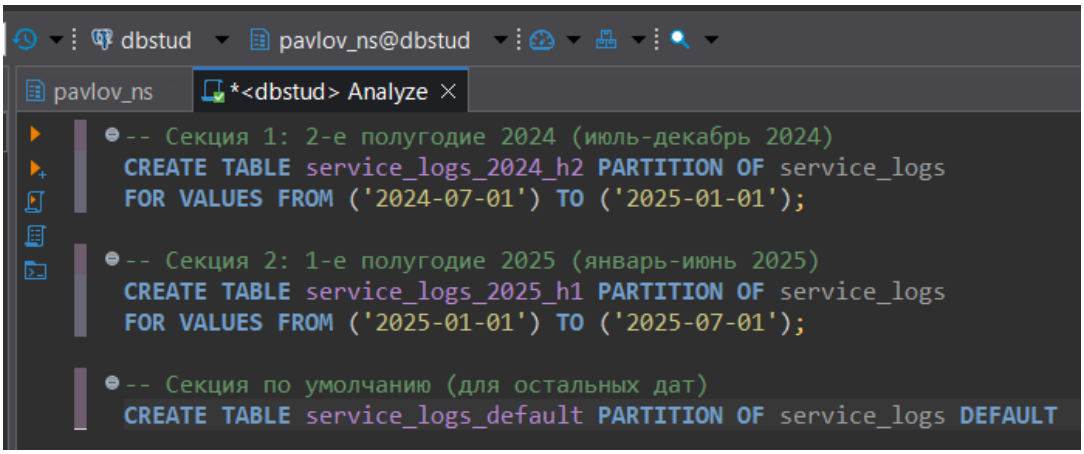

Рисунок 6 – Создание секций

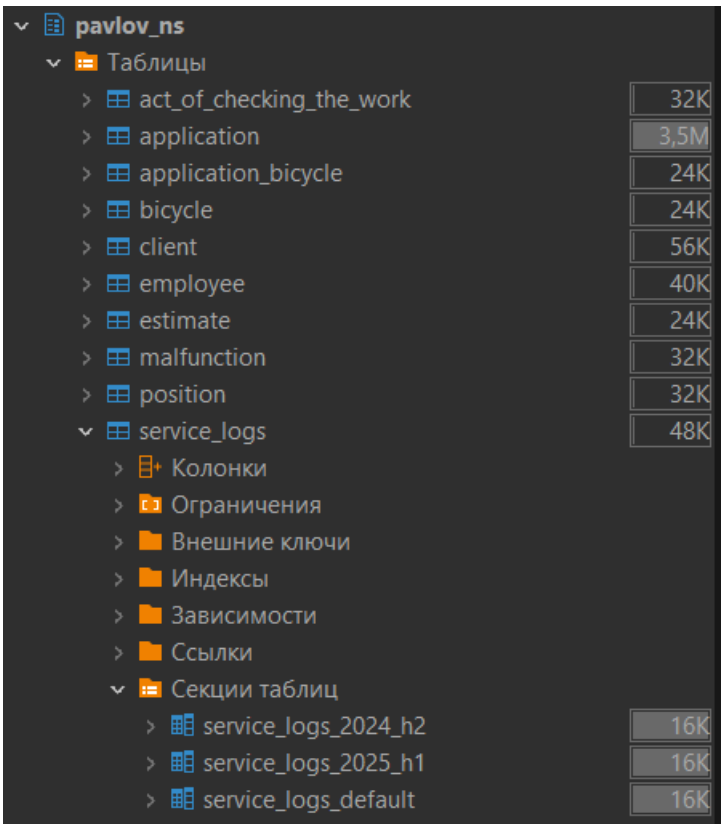

Рисунок 7 – Структура таблиц в навигаторе объектов (видна иерархия)

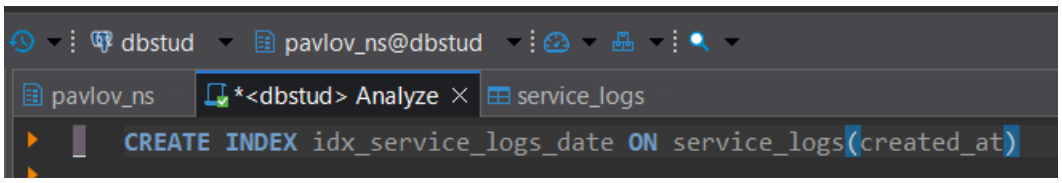

Рисунок 8 – Создание индекса

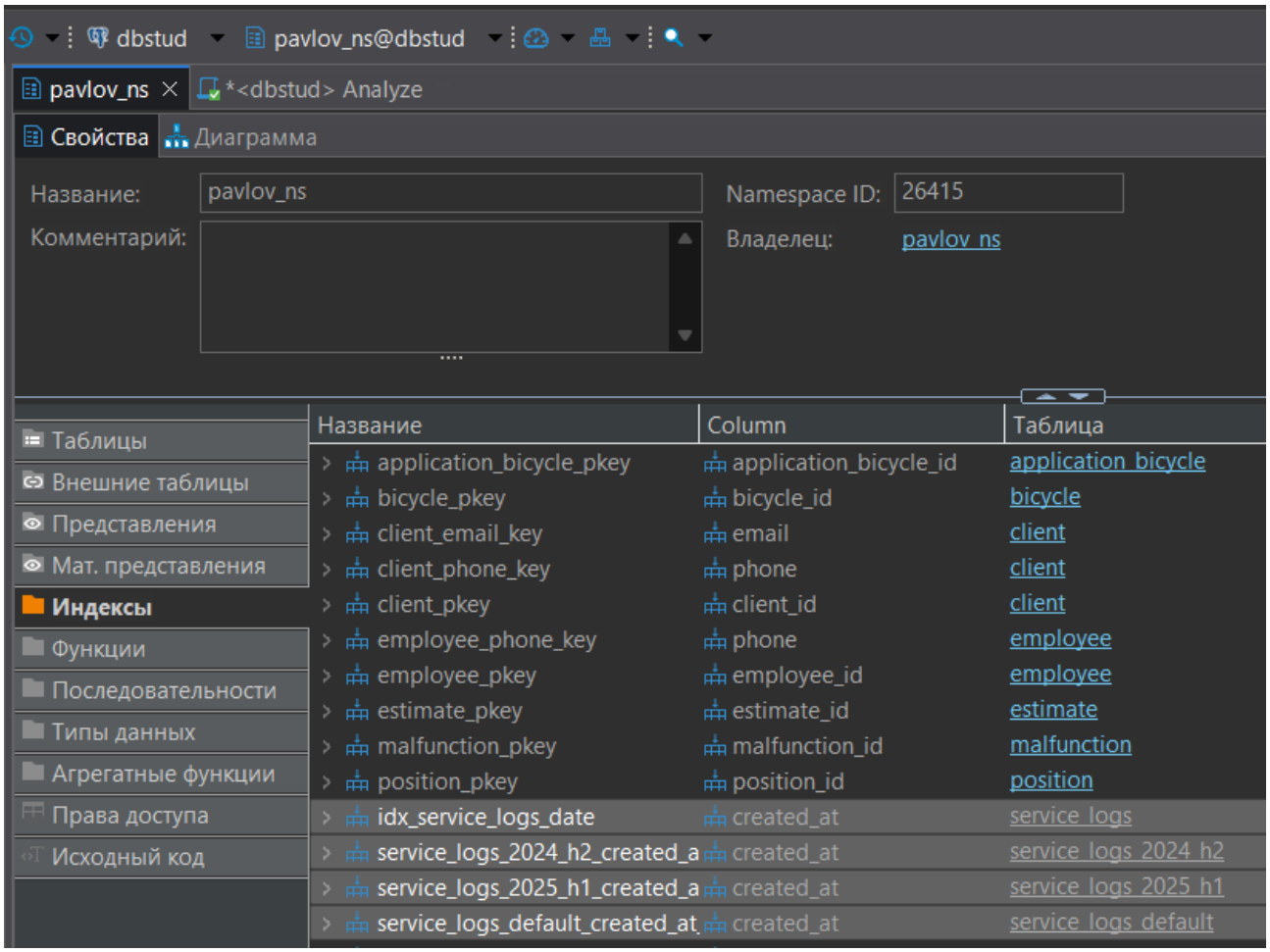

Рисунок 9 – Проверка применения индекса ко всем секциям

2.4 ЗАДАНИЕ 3. ГЕНЕРАЦИЯ ДАННЫХ

```
DO $$
DECLARE
    i INT;
    random_date TIMESTAMPTZ;
    random_employee INT;
    event_type TEXT;
    json_payload JSONB;
BEGIN
    FOR i IN 1..20000 LOOP
        random_date := '2024-07-01'::TIMESTAMPTZ + (random() * 365 * INTERVAL '1 day');

        random_employee := floor(random() * 10 + 1)::INT;

        IF (i % 2 = 0) THEN
            event_type := 'payment';
            json_payload := jsonb_build_object(
                'event', event_type,
                'details', jsonb_build_object(
                    'order_id', i,
                    'amount', floor(random() * 80000 + 500),
                    'method', CASE floor(random() * 2)::INT
                        WHEN 0 THEN 'Наличные'
                        ELSE 'Карта'
                    END
                )
            );
        ELSE
            event_type := 'login';
            json_payload := jsonb_build_object(
                'event', event_type,
                'ip', '192.168.1.' || floor(random() * 255)::TEXT,
                'success', (random() > 0.1)
            );
        END IF;

        INSERT INTO service_logs (created_at, employee_id, event_type, event_data)
        VALUES (random_date, random_employee, event_type, json_payload);
    END LOOP;
END $$
```

Рисунок 10 – Скрипт генерации данных

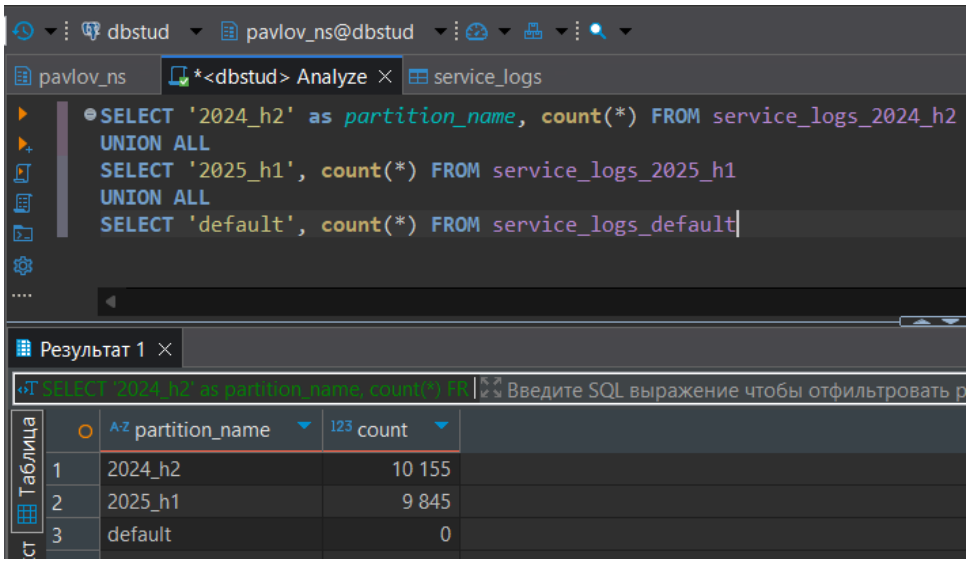

Рисунок 11 – Проверка количества записей в секциях

2.5 ЗАДАНИЕ 4. АНАЛИЗ И ПОИСК ПО JSONB

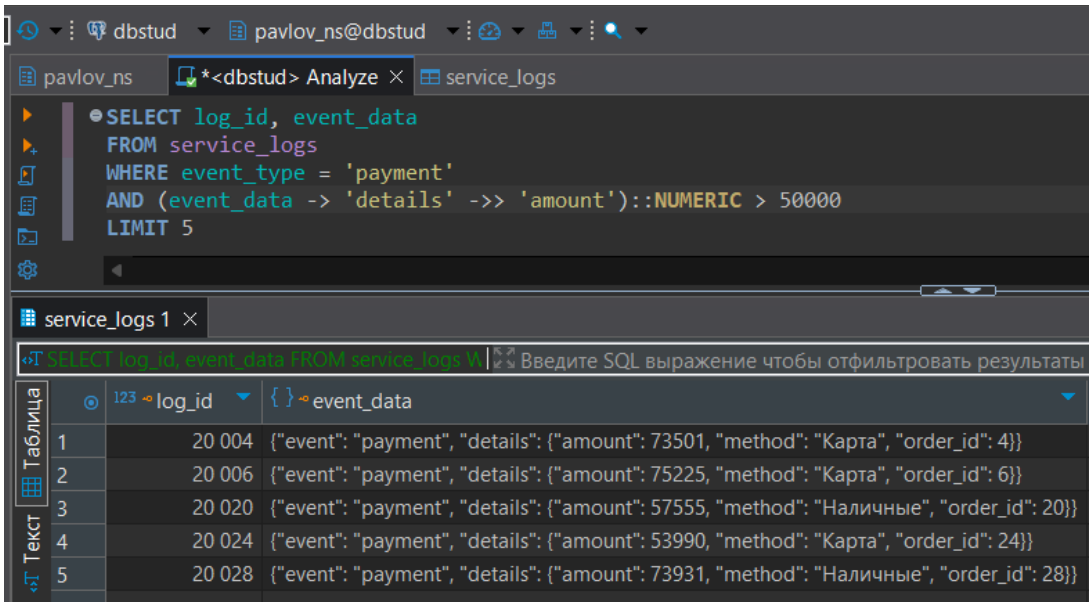

Рисунок 12 – Фильтрация со сложным условием

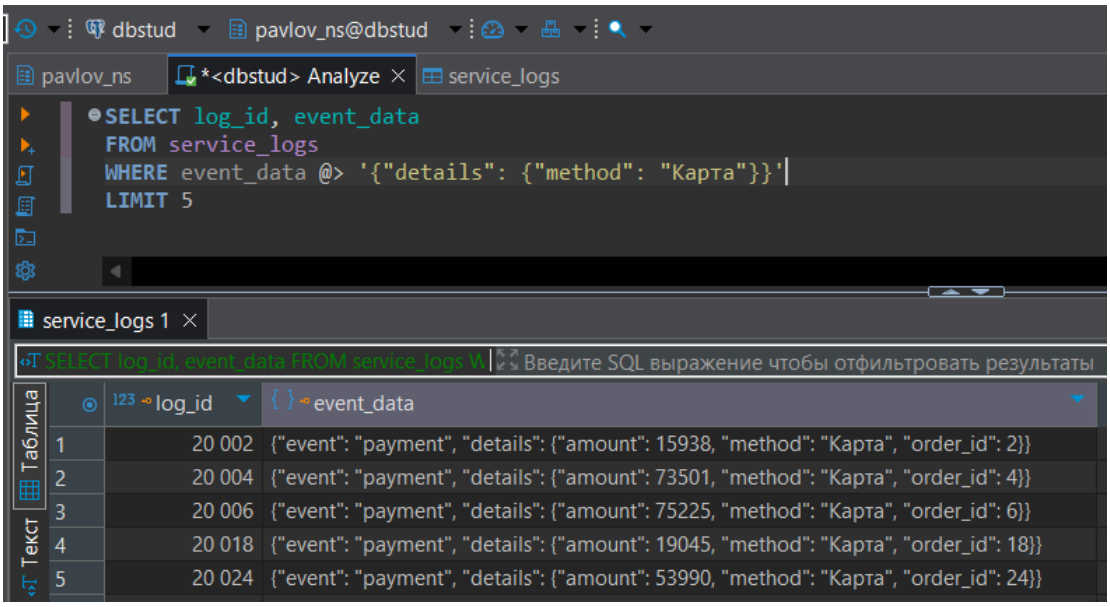

Рисунок 13 – Поиск по вхождению фрагмента

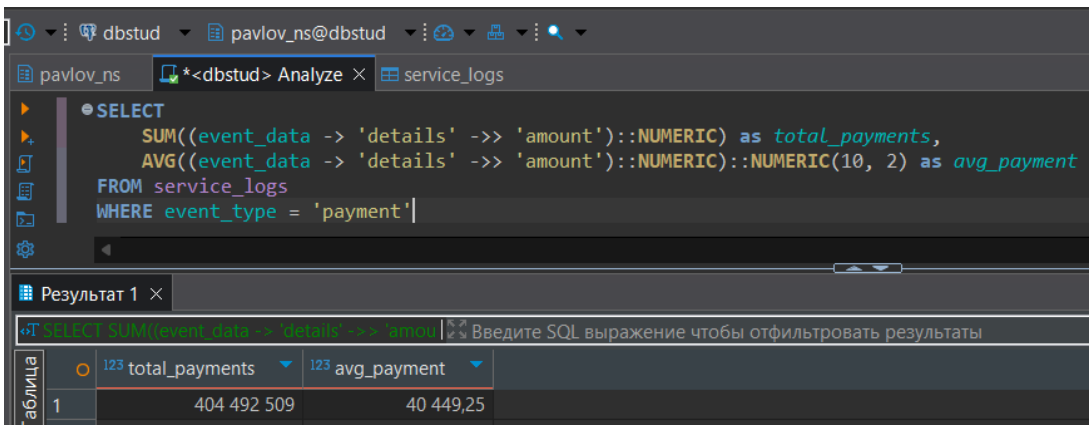

Рисунок 14 – Агрегация данных

2.6 ЗАДАНИЕ 5. МОДИФИКАЦИЯ ДАННЫХ JSONB

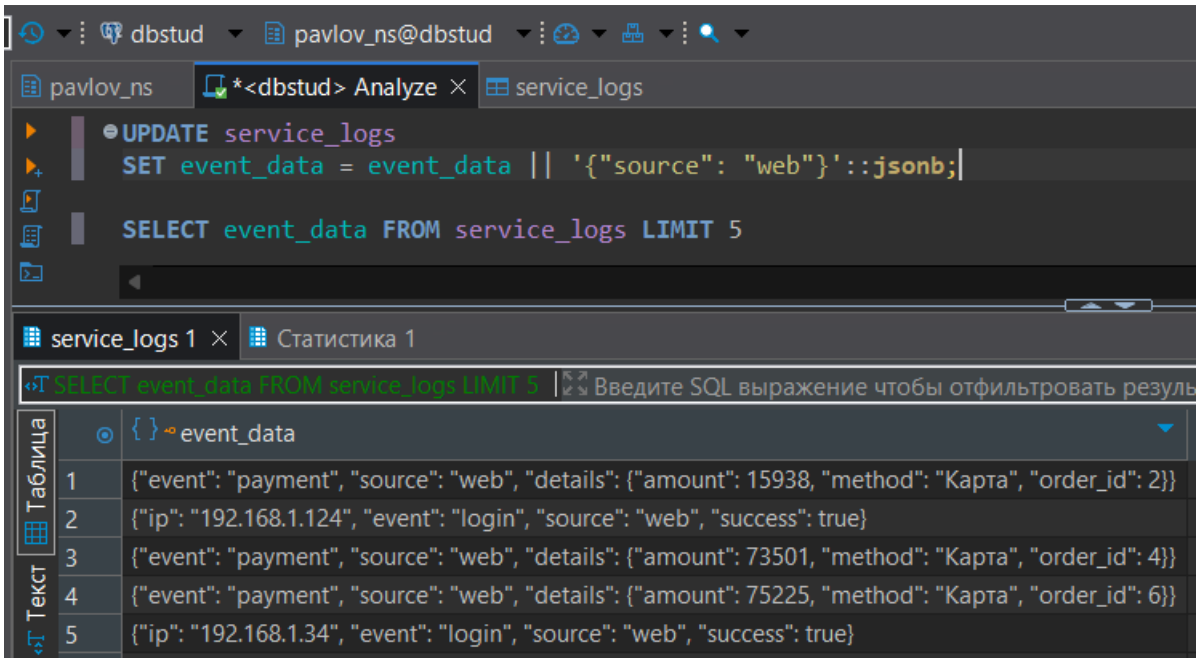

Рисунок 15 – Массовое обновление

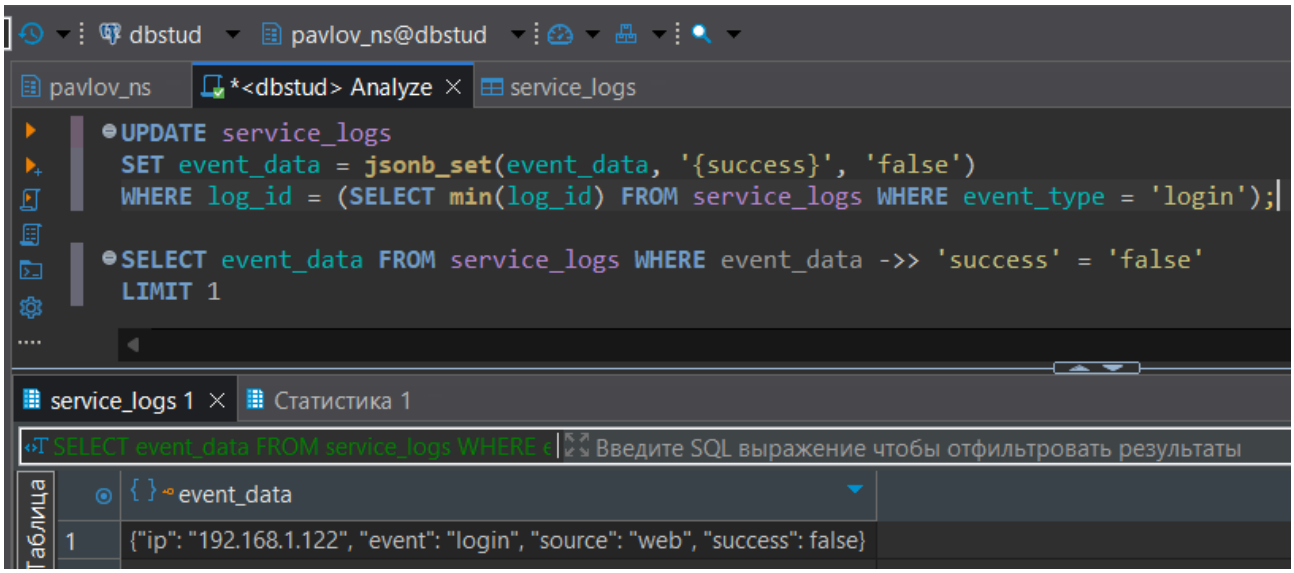

Рисунок 16 – Использование jsonb_set