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

по дисциплине «Разработка баз данных»

Практическое занятие №8

Студенты группы *ИКБО-65-23 Олефиров Г.Г.*

_____
(подпись)

Ассистент Морозов Д.В.

_____
(подпись)

Отчет представлен «___»_____2025 г.

Москва 2025 г.

ПРАКТИЧЕСКАЯ РАБОТА №8. СЛОЖНЫЕ ТИПЫ ДАННЫХ И МАСШТАБИРОВАНИЕ POSTGRESQL

Постановка задачи:
Целью данной практической работы является 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.

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).**1. Аудит и логирование.**
Запись факта изменения данных.
Пример: при любом изменении столбца salary в таблице employees создавать запись в salary_log, сохраняя OLD.salary, NEW.salary и CURRENT_USER.

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 таблицамсекциям.

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.

1.Работа с бинарными данными(BYTEA) и хеширование

1.1 Создание таблиц

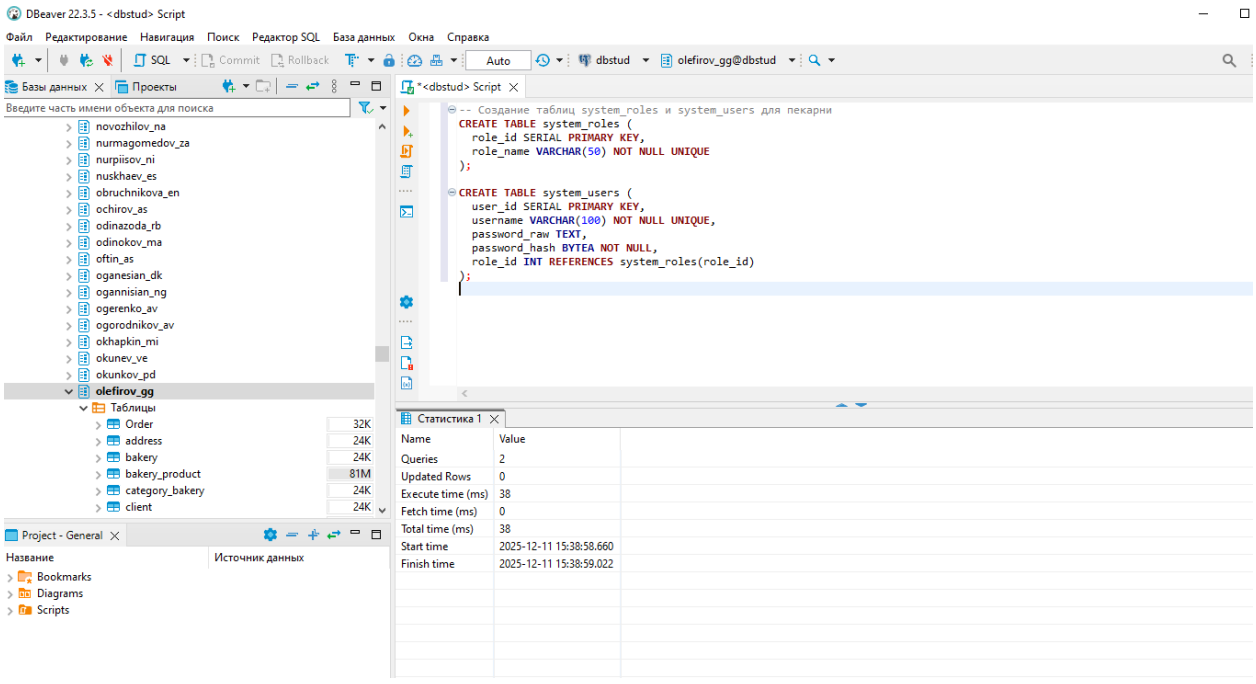

Рисунок 1 – Создание таблиц

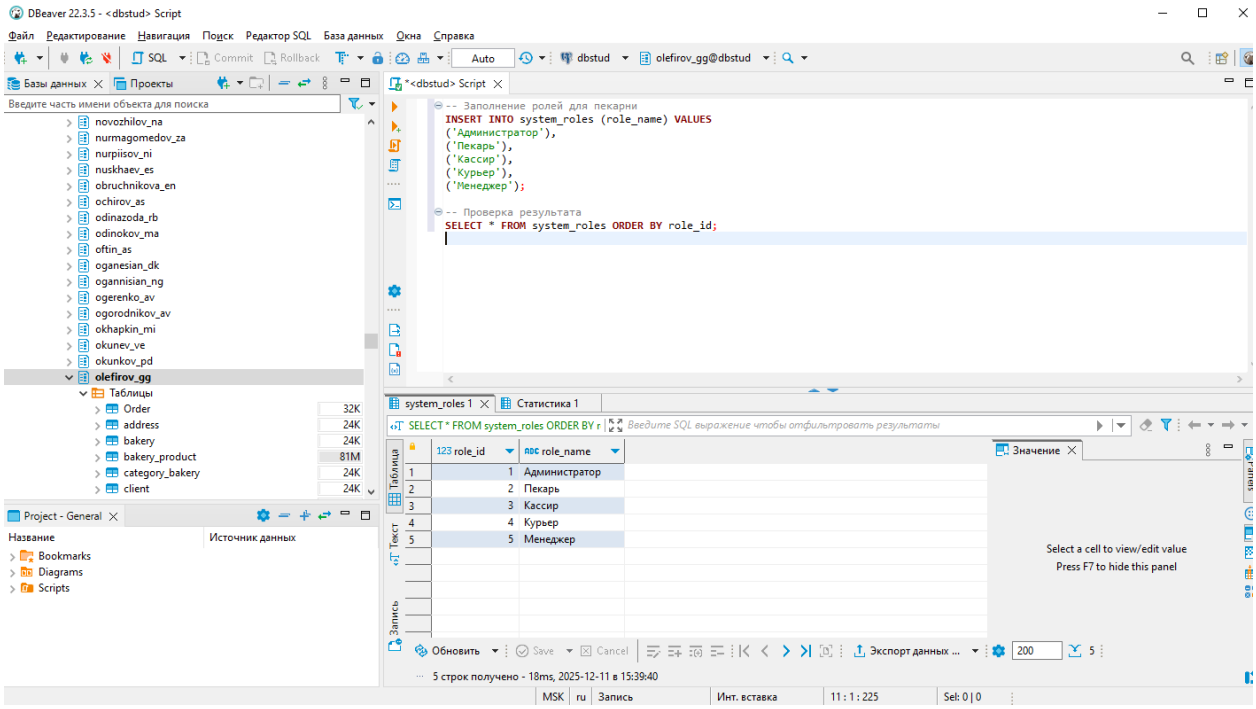

Рисунок 2 – Заполнение ролей

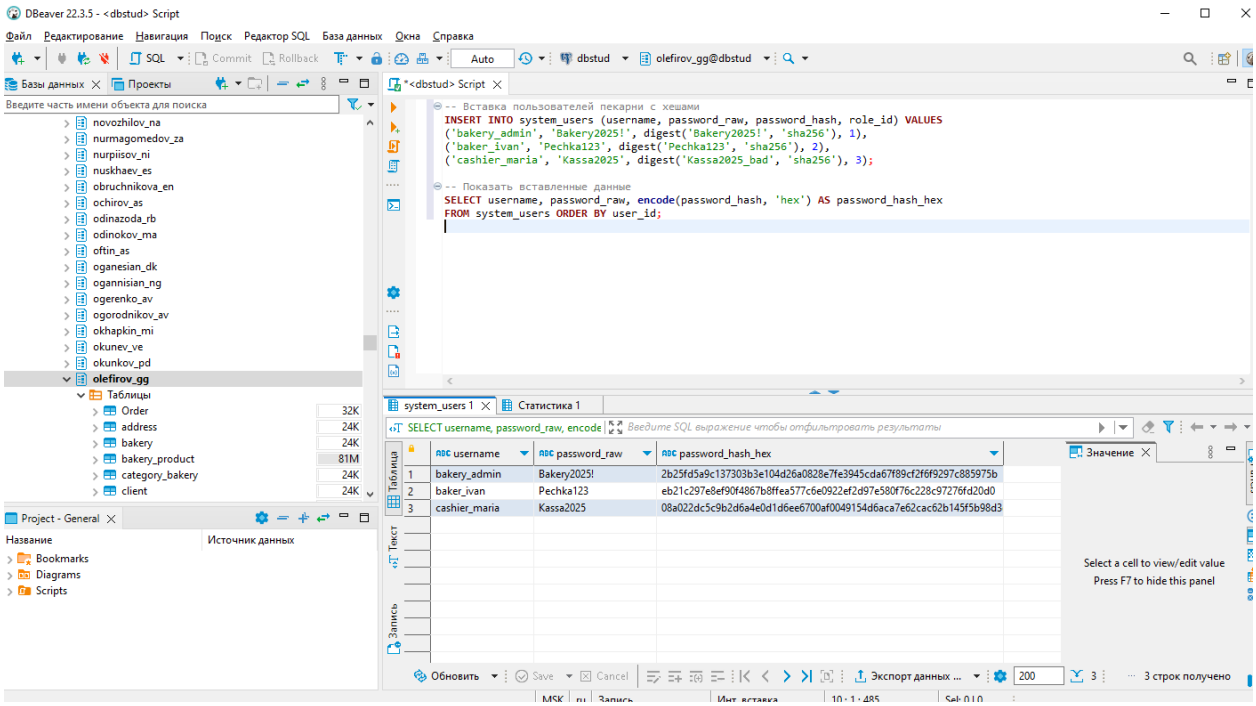

Рисунок 3 – Хеширование паролей и вставление пользователей

1.2 Проверка паролей

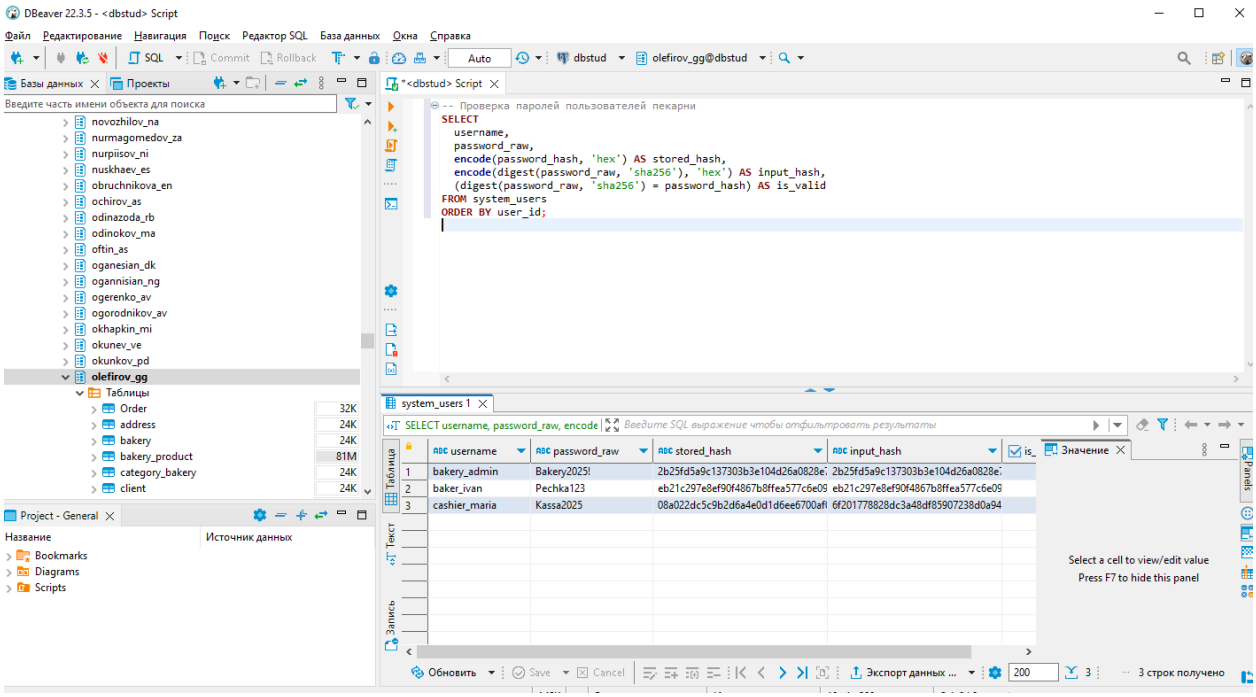

Рисунок 4 – Проверка соответствия паролей

2.Масштабирование базы данных(секционирование)

2.1 Создание таблицы логов

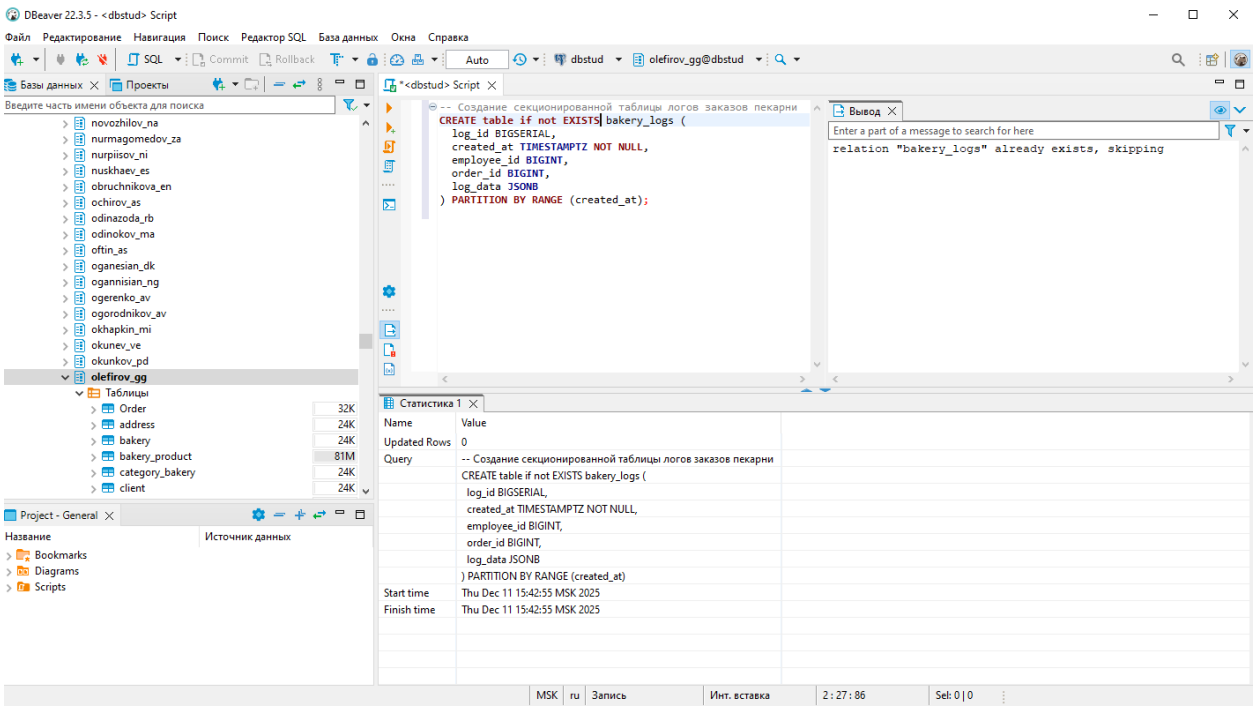

Рисунок 5 – Создание таблицы логов

2.2 Создание секций(партиций)

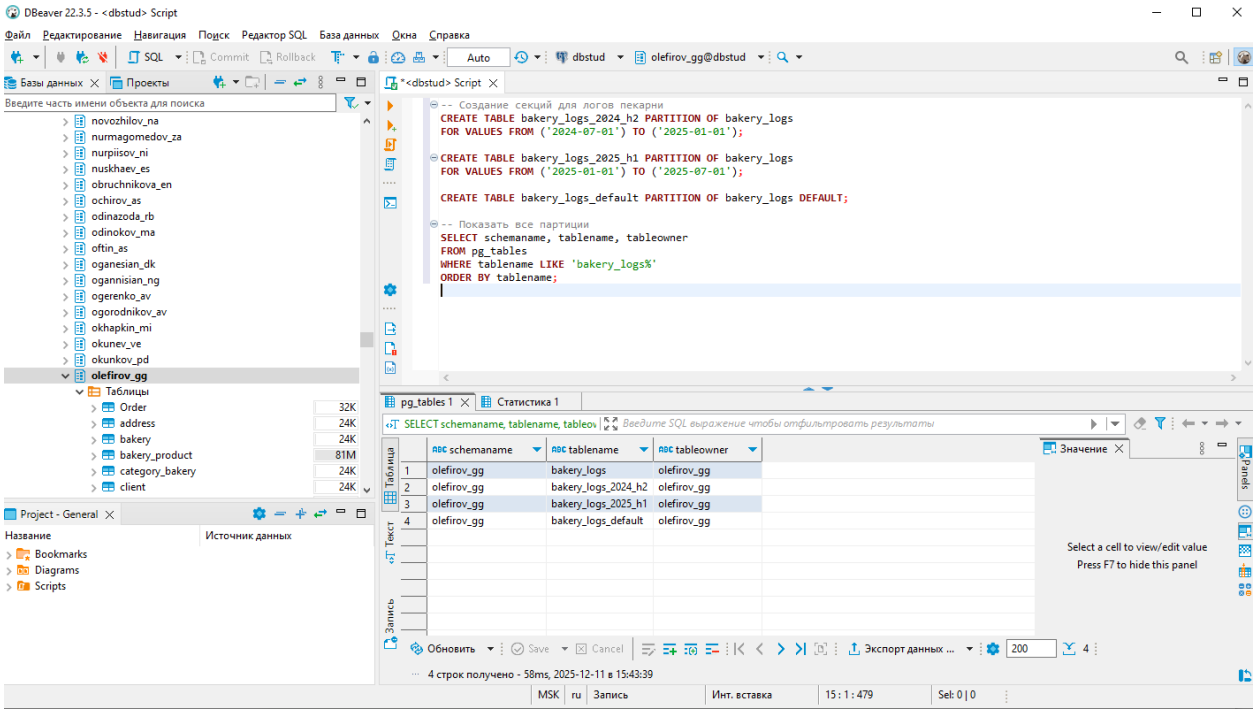

Рисунок 6 – Создание секций

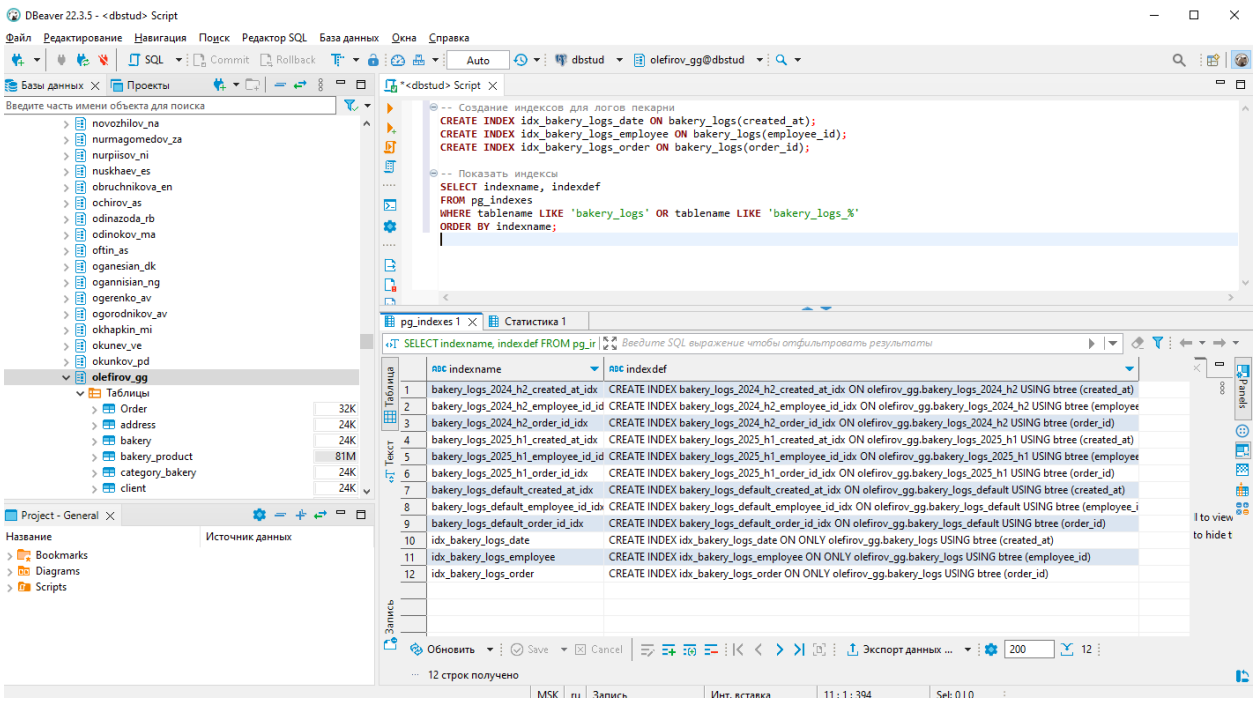

Рисунок 7 – Создание индексов

3. Генерация данных и тип JSONB

3.1 Генерация тестовых данных

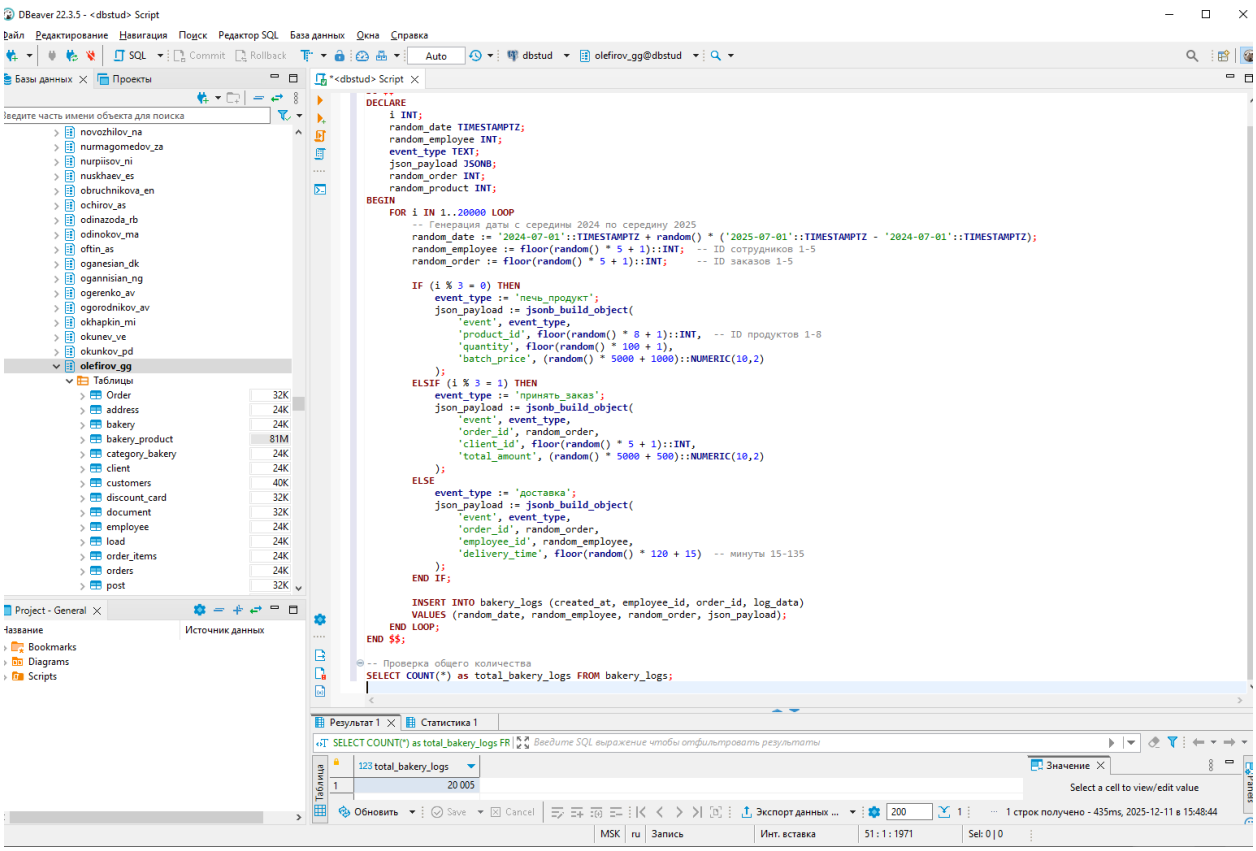

Рисунок 8 – Скрипт генерации данных

3.2 Проверка распределения данных

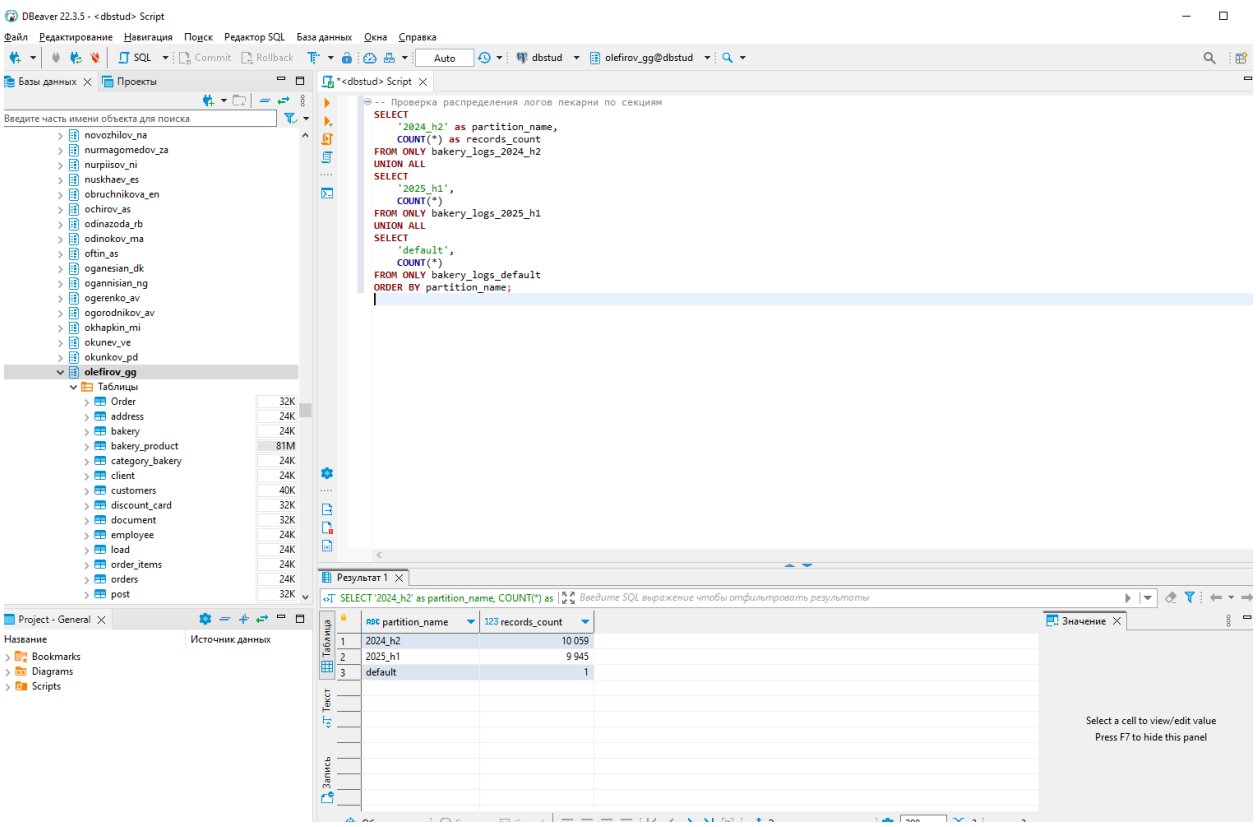

Рисунок 9 – Результат распределения данных по секциям

4. Анализ данных JSONB

4.1 Поиск по ключу и фильтрация с условием

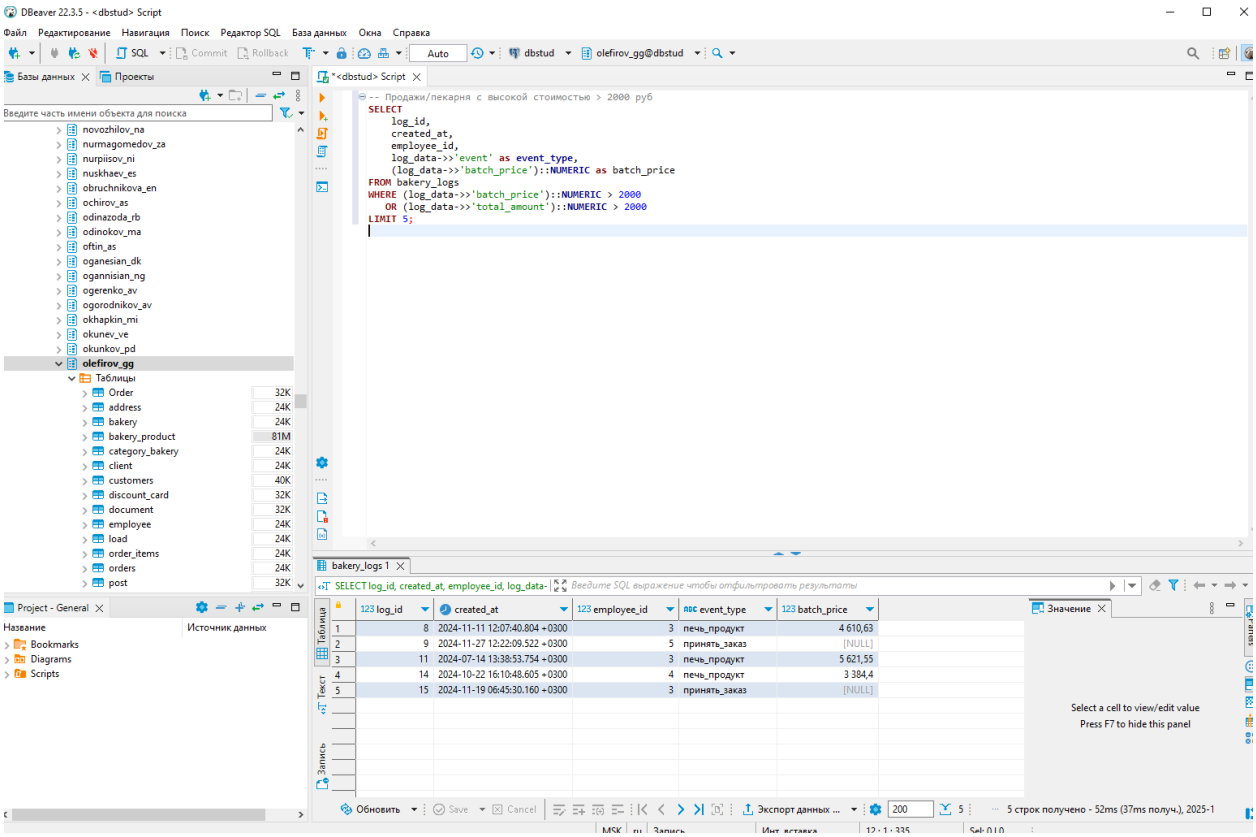

Рисунок 10 – Результат выборки суммы > 2000

4.2 Фильтрация по вхождению (JSONB Containment)

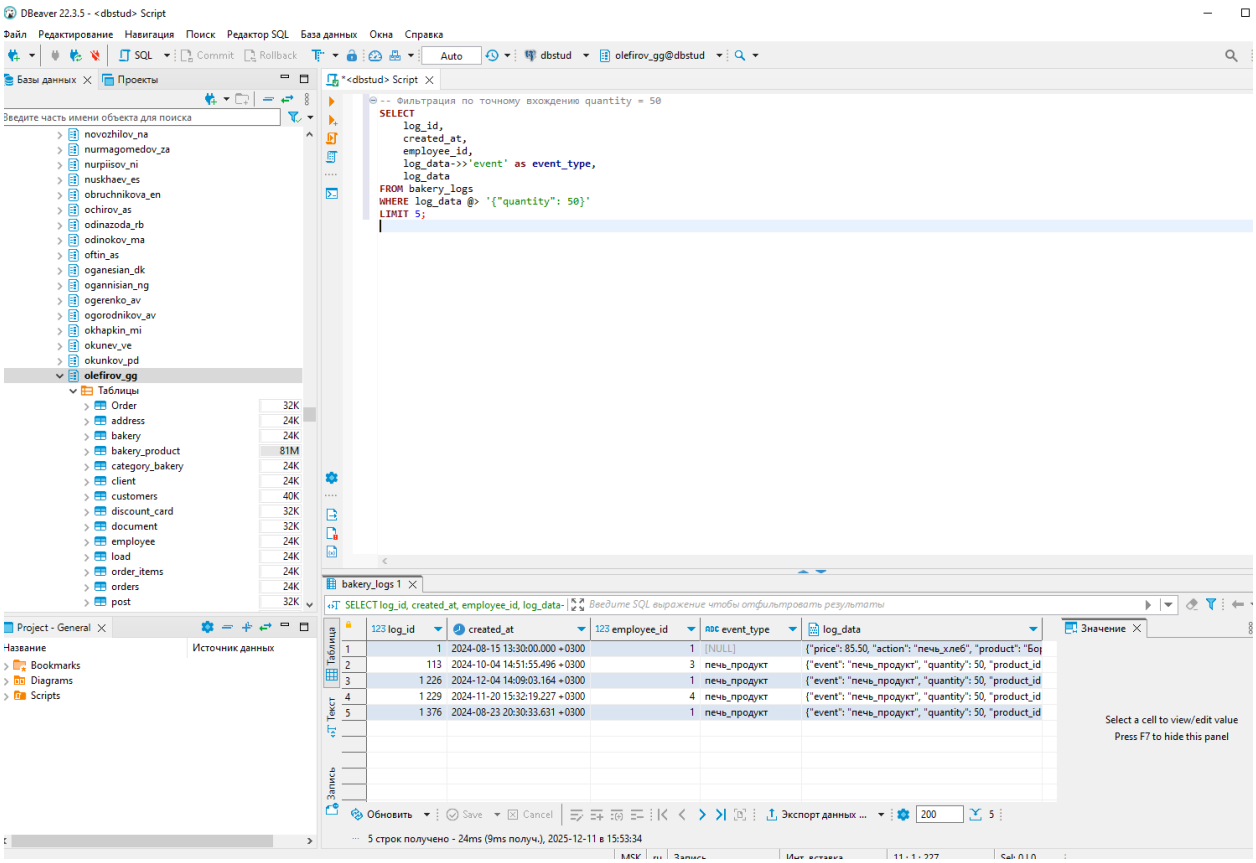

Рисунок 11 – Результаты поиска всех записей равный quantity = 50

4.3 Агрегация данных

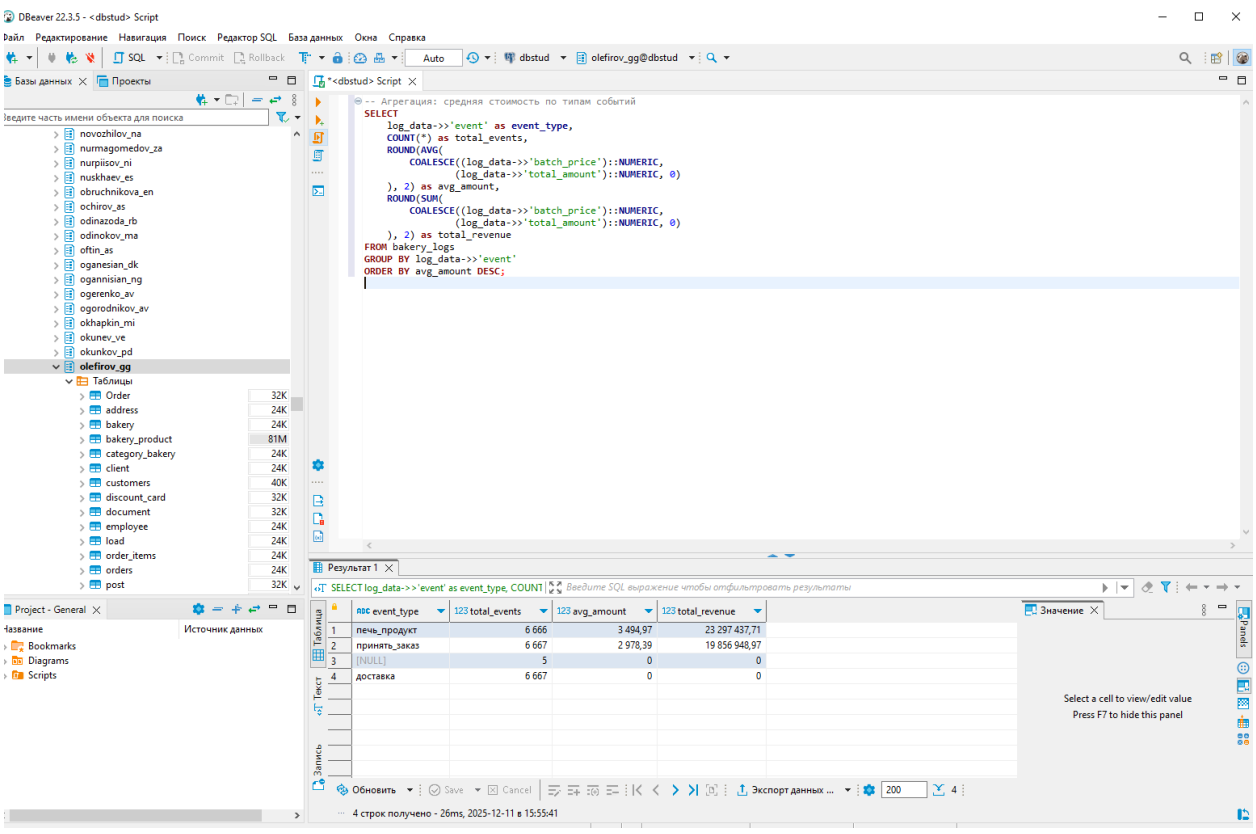

Рисунок 12 – Средняя стоимость операций пекарни

5. Модификация данных JSONB

5.1 Массовое обновление

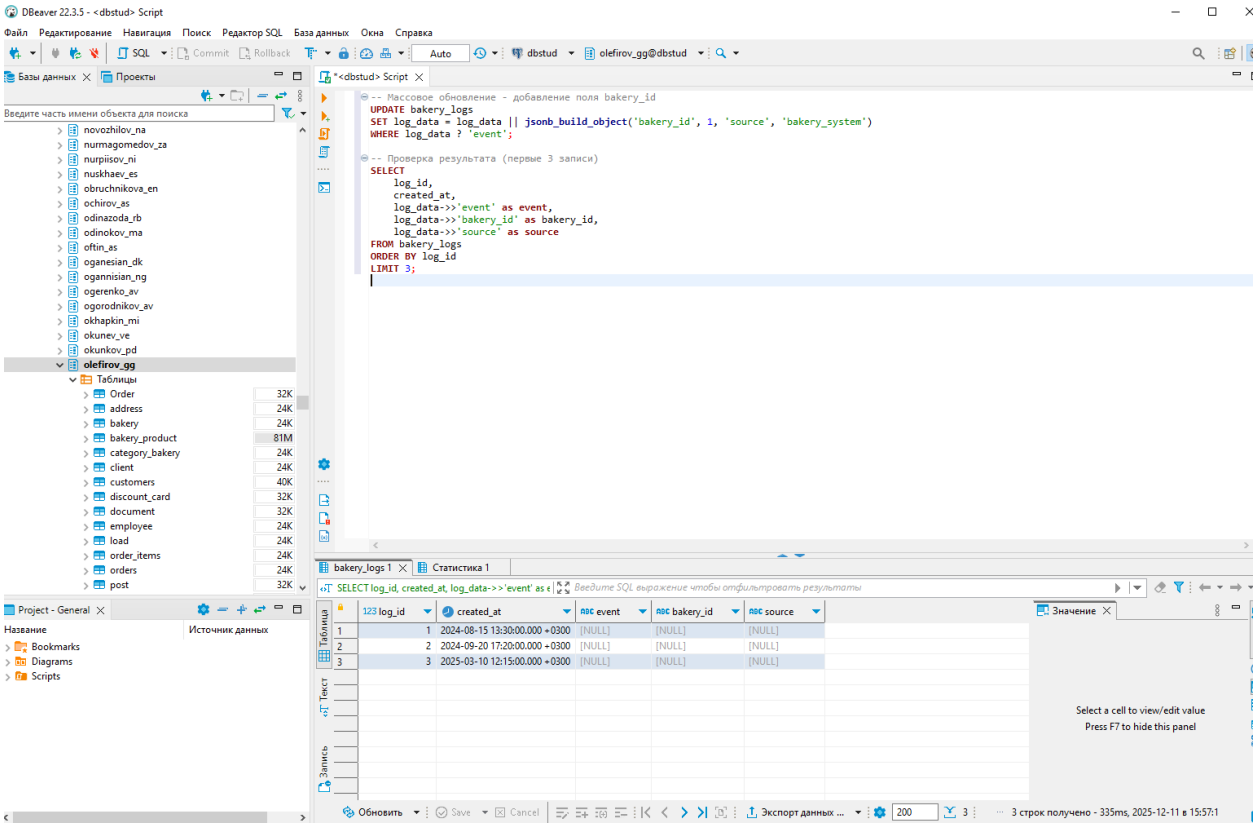

Рисунок 13 – Результат добавления поля bakery_id

5.2 Точечное изменение (функция jsonb_set)

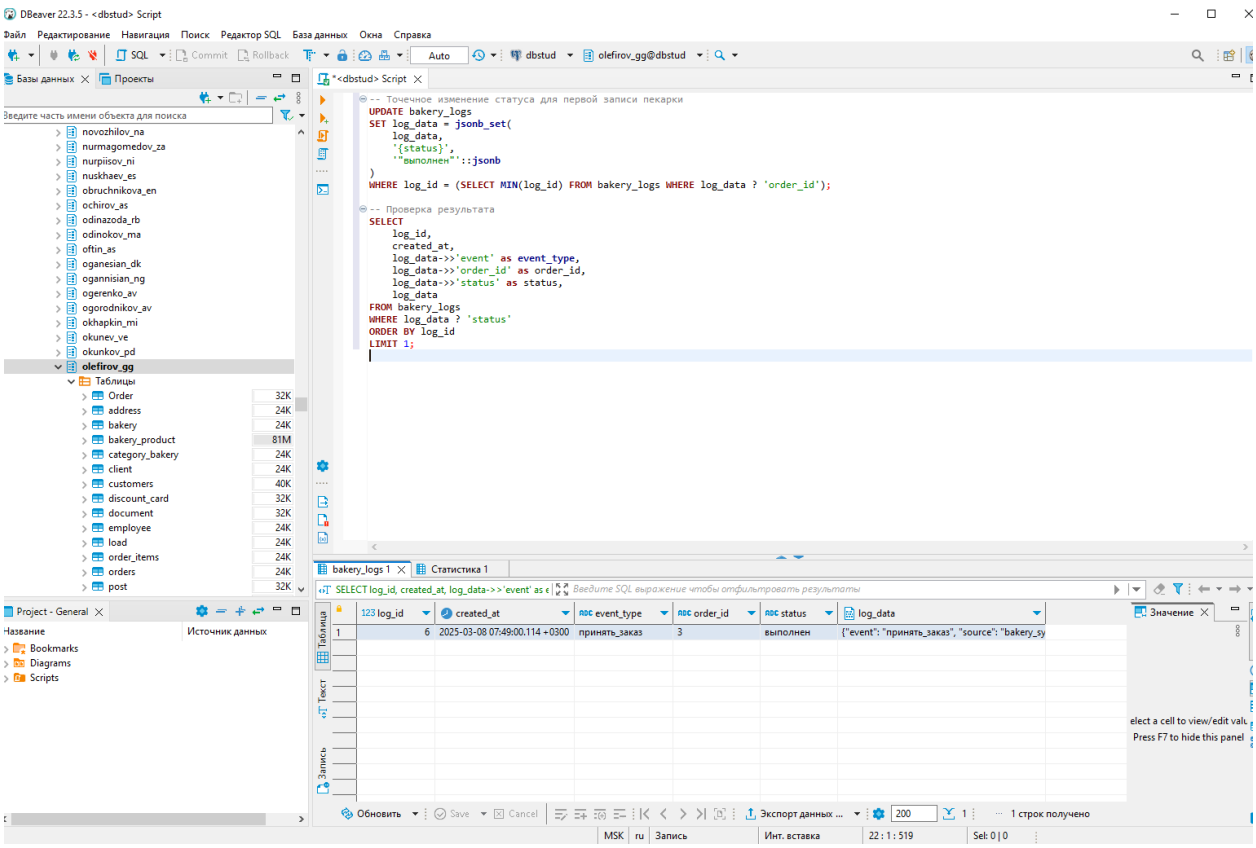

Рисунок 14 – Результат изменения статуса