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 №7

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: формирование у студентов практических навыков анализа и оптимизации производительности SQL-запросов, а также освоение механизмов управления транзакциями для обеспечения целостности данных (согласно принципам ACID) в СУБД PostgreSQL.

Задачи:

Задание №1: анализ и оптимизация (3 сценария)

Определить три различных «медленных» запроса к вашей БД, которые можно оптимизировать с помощью разных типов индексов (например, стандартный B-Tree, индекс по выражению, частичный/отфильтрованный индекс).

Если в вашей базе недостаточно данных, выполните для одной из своих таблиц действия по автоматической генерации содержимого, описанные в разделе «Подготовка базы данных».

Для каждого из 3-х сценариев:

1. Выполнить анализ запроса «КАК ЕСТЬ» (без индекса) с помощью EXPLAIN ANALYZE.
2. Привести план выполнения «ДО», письменно проанализировать его и выявить причину низкой производительности (например, Seq Scan).
3. Создать необходимый INDEX для оптимизации.
4. Повторно выполнить EXPLAIN ANALYZE.
5. Привести план выполнения «ПОСЛЕ», демонстрирующий использование индекса (например, Index Scan).
6. Обязательно сформировать сравнительную таблицу (см. Таблица 1), демонстрирующую разницу в производительности (план, Execution Time) «ДО» и «ПОСЛЕ».

Задание №2: демонстрация атомарной транзакции (COMMIT).

По примеру раздела 2.2 реализуйте в своей базе одну бизнес-операцию (минимум две связанные операции изменения данных) внутри транзакции BEGIN…COMMIT.

Задокументировать все шаги и результаты, сделать выводы.

Задание №3: демонстрация отката транзакции (ROLLBACK).

Адаптировать приведённый в разделе 2.3 SQL-скрипт, моделирующий сбой операции, под свою предметную область, повторив описанные действия.

Задокументировать все шаги и результаты, сделать выводы.

Задание №4: моделирование аномалии «Неповторяемое чтение».

Используя два редактора SQL, смоделировать проблему «неповторяемое чтение» на уровне изоляции по умолчанию (READ COMMITTED) по приведённому в разделе 2.4 образцу.

Задокументировать все шаги и результаты, сделать выводы.

Задание №5: устранение аномалии «Неповторяемое чтение».

Повторить моделирование из Задания №4, но с использованием уровня изоляции REPEATABLE READ. В качестве образца использовать раздел 2.5.

Задокументировать все шаги и результаты, сделать выводы.

2 ВЫПОЛНЕНИЕ РАБОТЫ

2.1 ИСХОДНЫЕ ТАБЛИЦЫ

application | Введите SQL выражение чтобы отфильтровать результаты

Таблица | Текст

	application_id	client_id	application_date	problem_description
1	1	1	2024-01-15	Прокол задней камеры, требуется замена
2	2	2	2024-01-16	Износ тормозных колодок, скрип при торможении
3	3	3	2024-01-17	Погнутый обод переднего колеса
4	4	4	2024-01-18	Проблемы с переключением передач
5	5	5	2024-01-19	Требуется полная регулировка трансмиссии
6	6	6	2024-01-20	Замена цепи и звезд
7	7	7	2024-01-21	Обслуживание вилки и амортизаторов
8	8	8	2024-01-22	Установка нового оборудования
9	9	9	2024-01-23	Диагностика электронной системы переключения
10	10	10	2024-01-24	Комплексное обслуживание после сезона
11	11	1	2024-02-01	Тестовый заказ

Рисунок 1 – Таблица application

user_order | Введите SQL выражение чтобы отфильтровать результаты

Таблица | Текст

	user_order_id	application_id	status_id	start_date	end_date
1	1	1	5	2024-01-15	2024-01-16
2	2	2	5	2024-01-16	2024-01-18
3	3	3	3	2024-02-17	2024-02-20
4	4	4	2	2024-07-18	2024-07-19
5	5	5	1	2024-07-19	2024-07-22
6	6	6	6	2024-11-20	2024-11-21
7	7	7	4	2025-01-21	2025-01-23
8	8	8	5	2025-01-22	2025-01-23
9	9	9	2	2025-01-23	2025-01-25
10	10	10	1	2025-01-24	2025-01-26
11	11	11	1	2024-02-01	2024-02-05

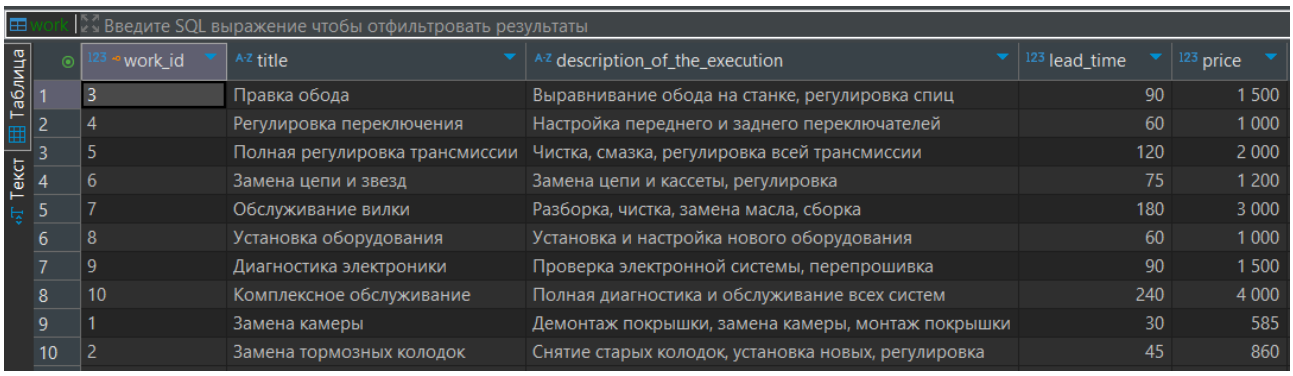
Рисунок 2 – Таблица user_order

transaction | Введите SQL выражение чтобы отфильтровать результаты

Таблица

	transaction_id	user_order_id	payment_date
1	1	1	2024-01-16
2	2	6	2024-01-21
3	3	8	2024-01-23

Рисунок 3 – Таблица transaction



work | Введите SQL выражение чтобы отфильтровать результаты

Таблица | Текст

	work_id	title	description_of_the_execution	lead_time	price
1	3	Правка обода	Выравнивание обода на станке, регулировка спиц	90	1 500
2	4	Регулировка переключения	Настройка переднего и заднего переключателей	60	1 000
3	5	Полная регулировка трансмиссии	Чистка, смазка, регулировка всей трансмиссии	120	2 000
4	6	Замена цепи и звезд	Замена цепи и кассеты, регулировка	75	1 200
5	7	Обслуживание вилки	Разборка, чистка, замена масла, сборка	180	3 000
6	8	Установка оборудования	Установка и настройка нового оборудования	60	1 000
7	9	Диагностика электроники	Проверка электронной системы, перепрошивка	90	1 500
8	10	Комплексное обслуживание	Полная диагностика и обслуживание всех систем	240	4 000
9	1	Замена камеры	Демонтаж покрышки, замена камеры, монтаж покрышки	30	585
10	2	Замена тормозных колодок	Снятие старых колодок, установка новых, регулировка	45	860

Рисунок 4 – Таблица work

2.2 ПОДГОТОВКА БД

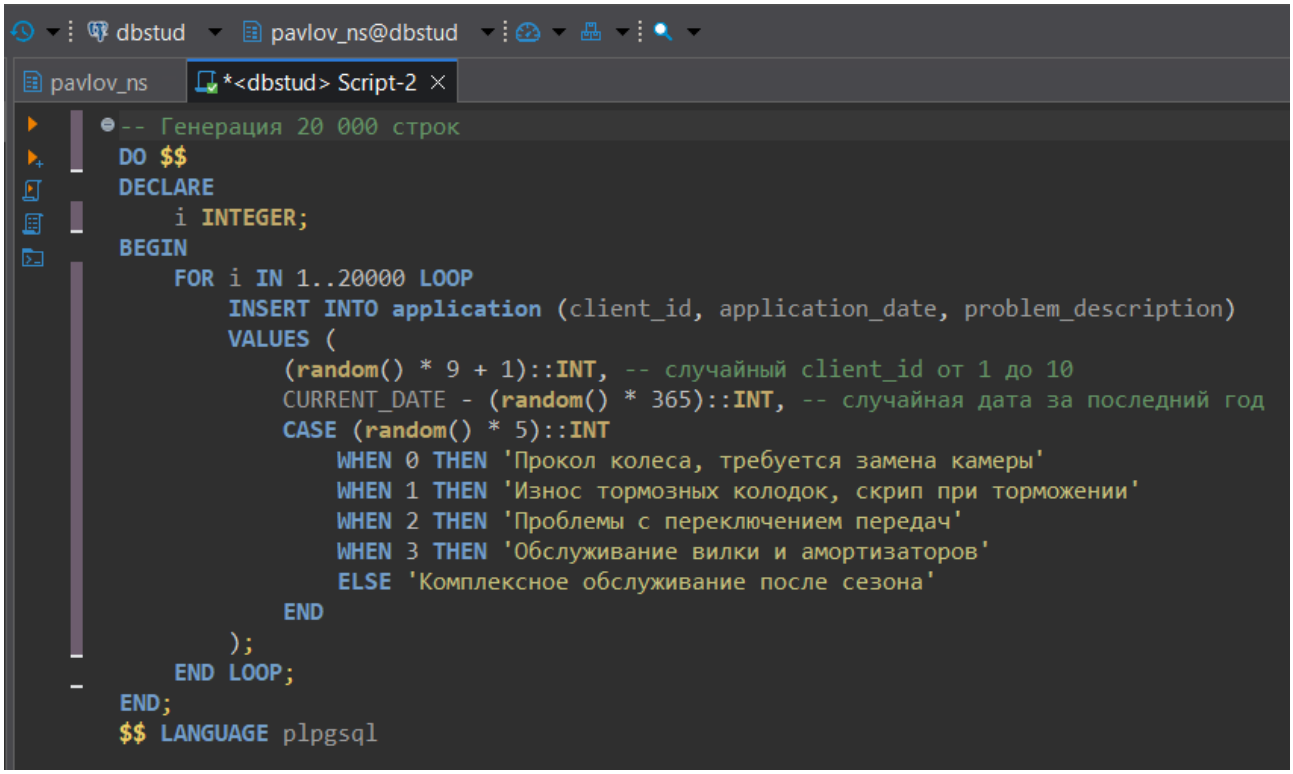


Рисунок 5 – Генерация 20000 строк в таблицу application для анализа

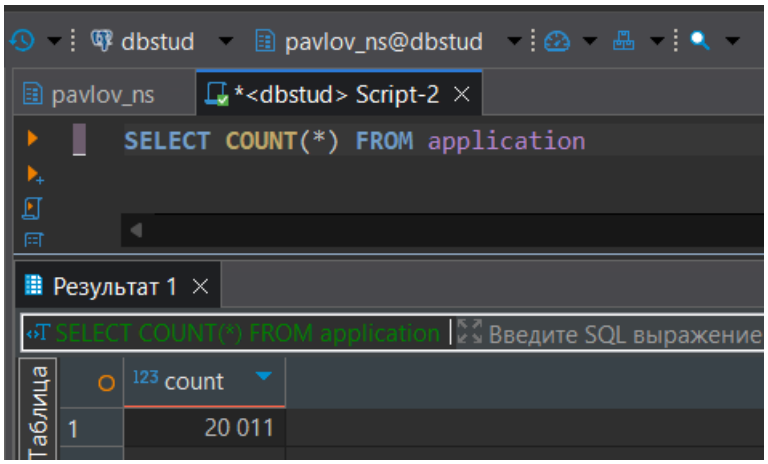


Рисунок 6 – Проверка добавления строк

2.3 ЗАДАНИЕ 1. АНАЛИЗ И ОПТИМИЗАЦИЯ

2.3.1 Сценарий 1. B-Tree

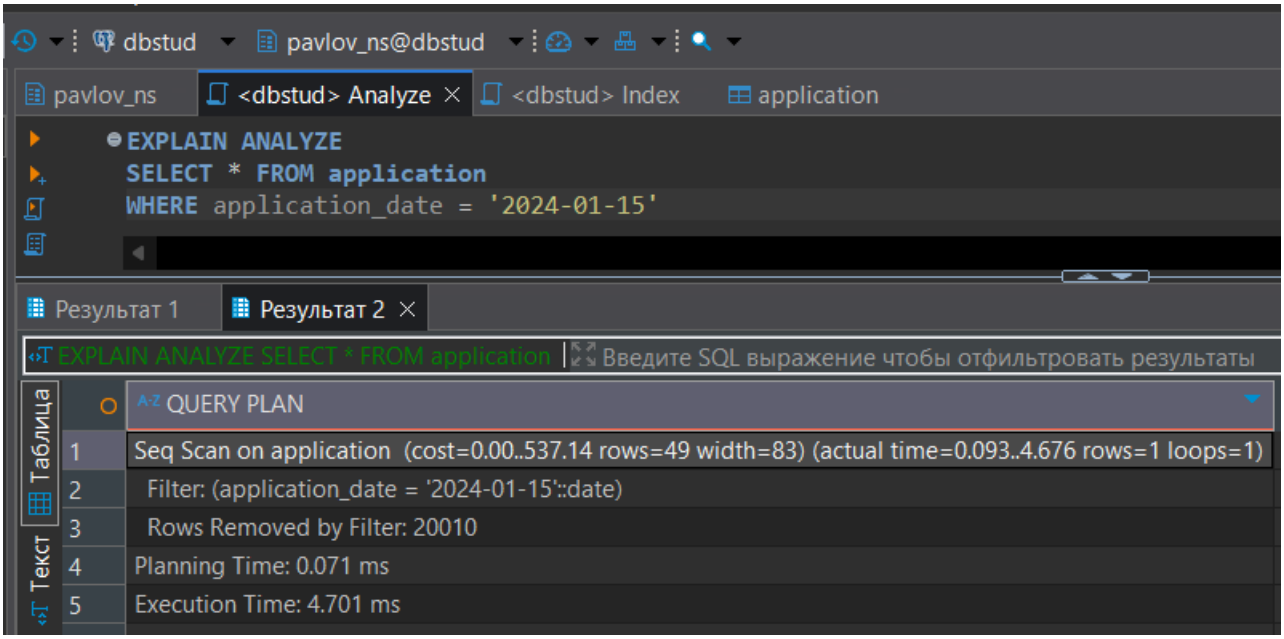


Рисунок 7 – Анализ запроса без индекса

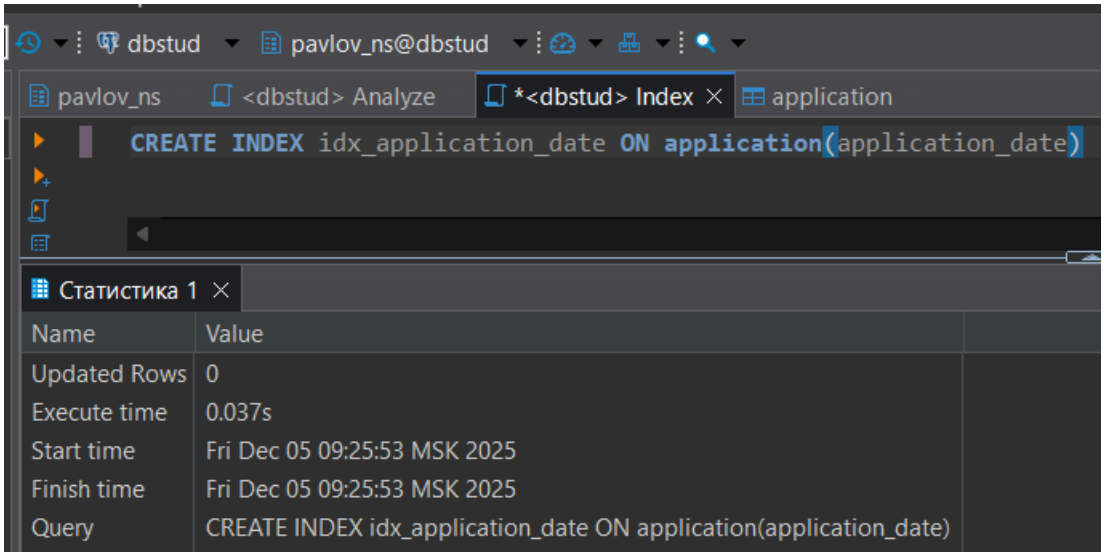


Рисунок 8 – Создание индекса дат для таблицы

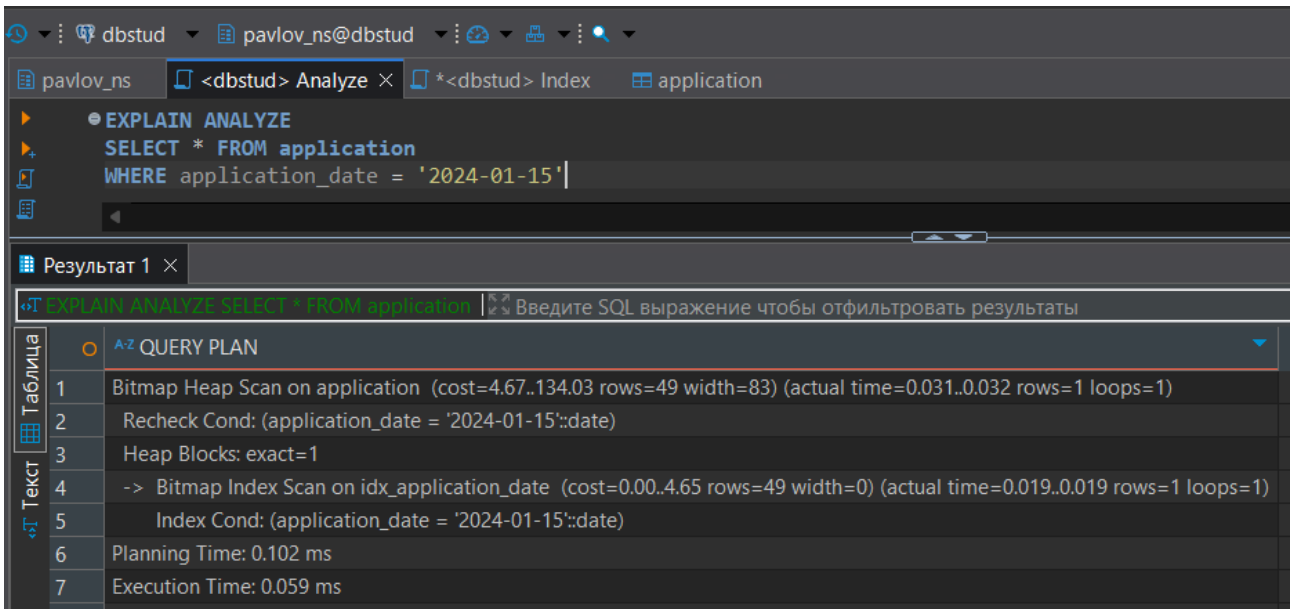


Рисунок 9 – Анализ запроса с созданным индексом

Таблица 1 – Сравнительная таблица времени выполнения

Метрика	До оптимизации	После оптимизации	Вывод
План (Оператор)	Seq Scan	Bitmap Index Scan	Выбор отличается
Execution Time	4.701 ms	0.059 ms	Запрос **ускорился** примерно в 80 раз

2.3.2 Сценарий 2. Индекс по выражению

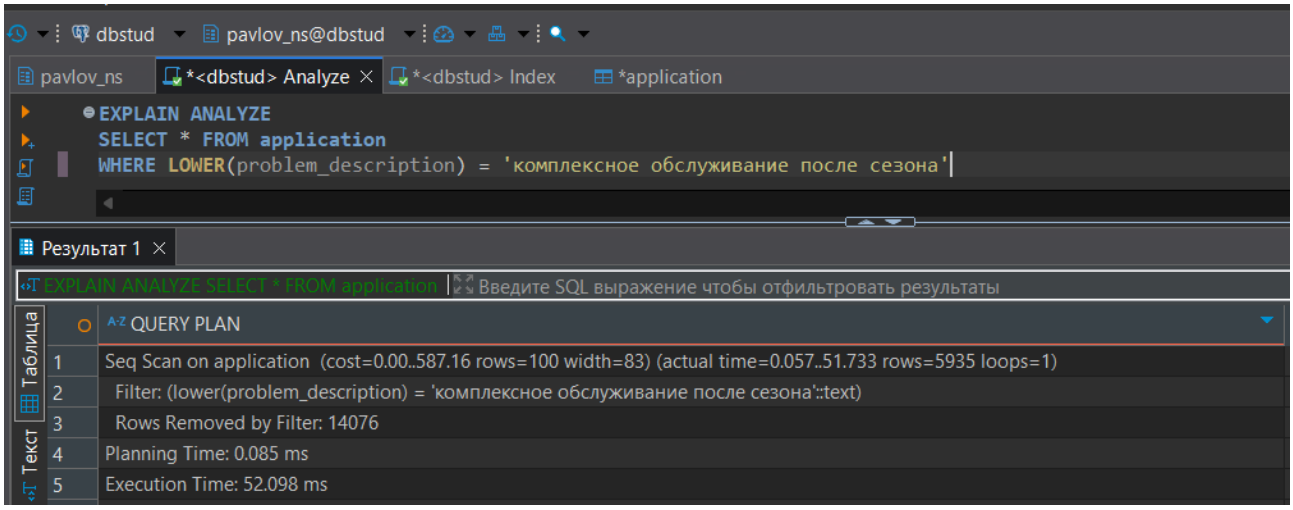


Рисунок 10 – Анализ запроса без индекса

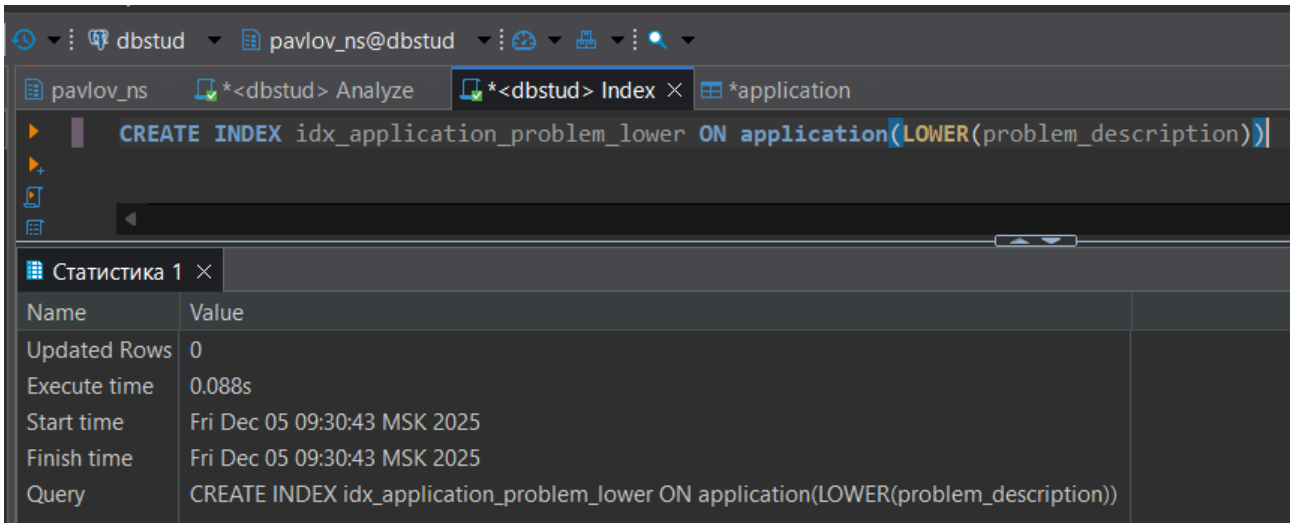


Рисунок 11 – Создание индекса по выражению LOWER()

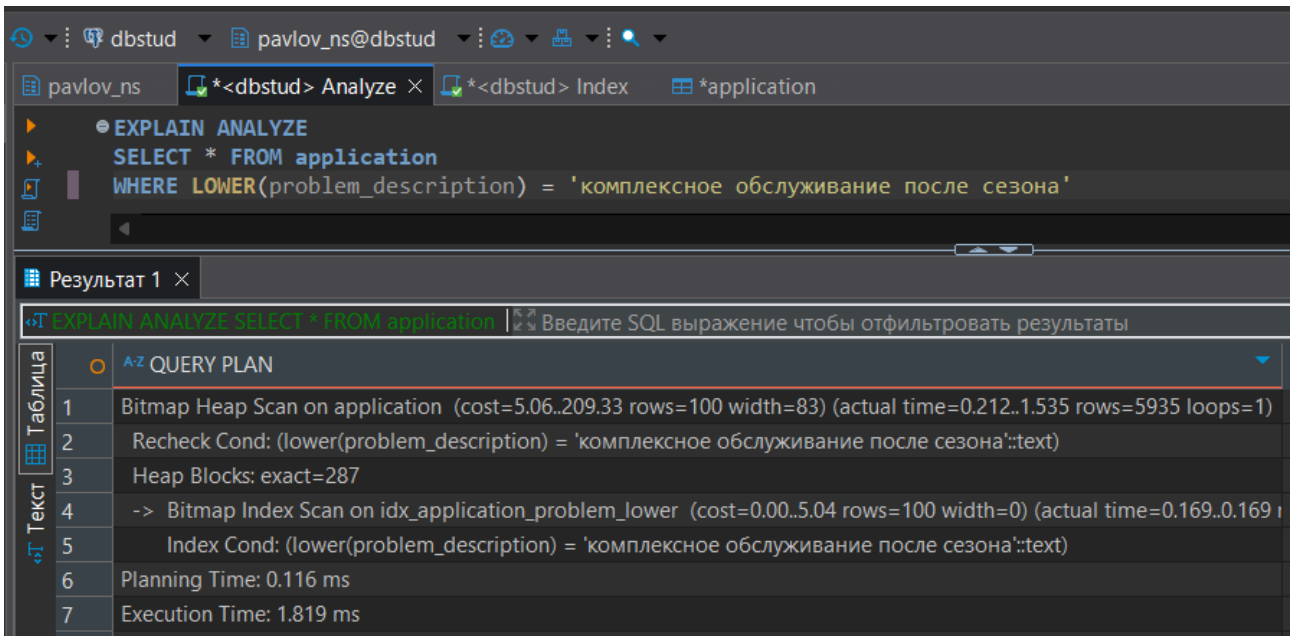


Рисунок 12 – Анализ запроса с использованием индекса

Таблица 2 – Сравнительная таблица времени выполнения

Метрика	**До оптимизации**	**После оптимизации**	**Вывод**
План (Оператор)	Seq Scan	Bitmap Index Scan	Выбор отличается
Execution Time	52.098 ms	1.819 ms	Запрос **ускорился** примерно в 28 раз

2.3.3 Сценарий 3. Частичный индекс

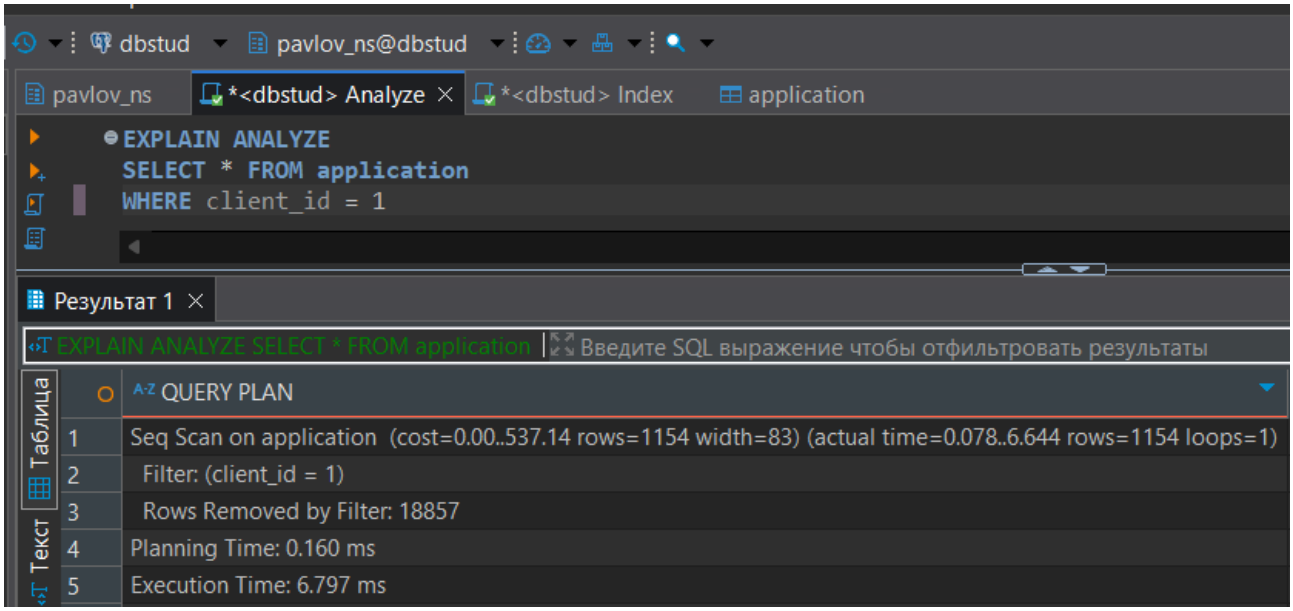


Рисунок 13 – Анализ запроса без индекса

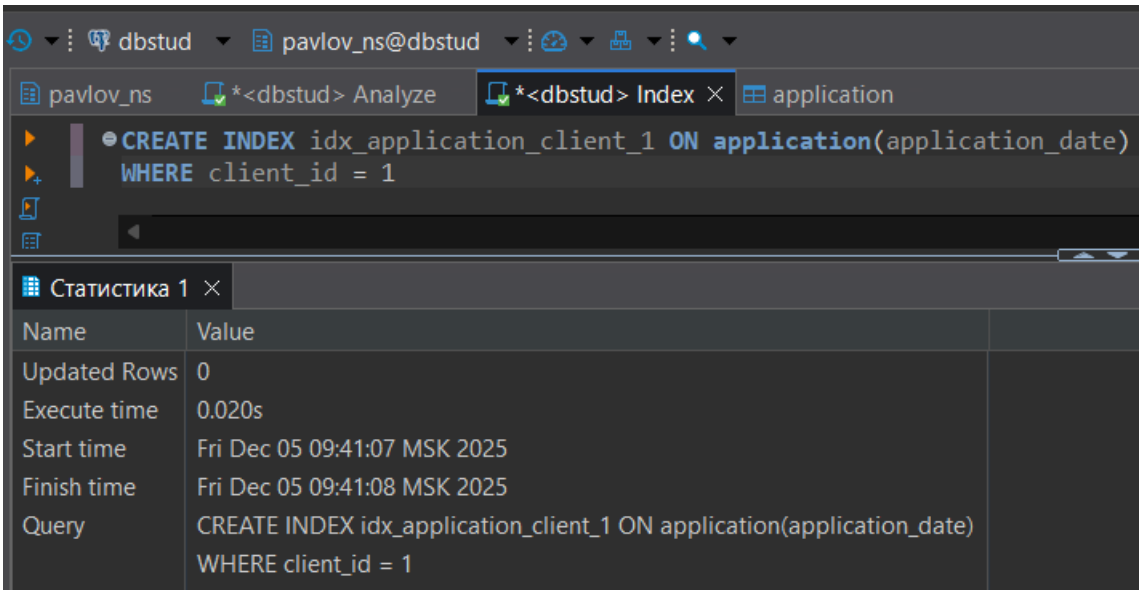


Рисунок 14 – Добавление частичного индекса

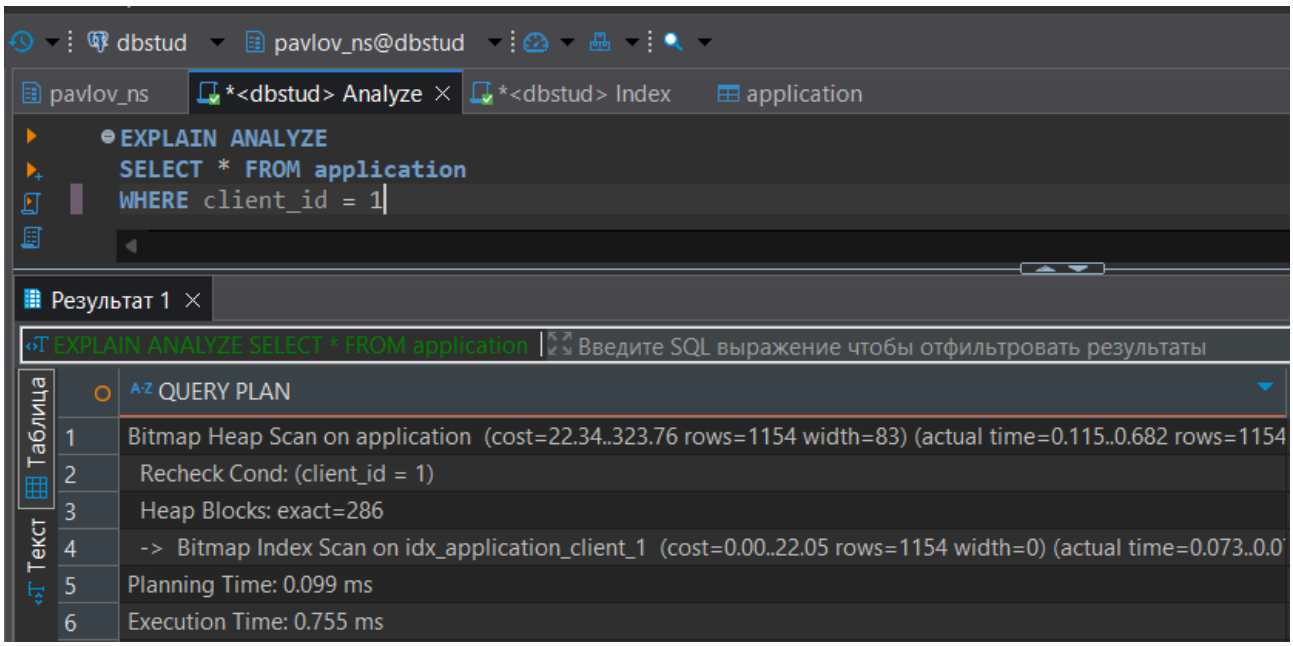


Рисунок 15 – Анализ запроса с использованием частичного индекса

Таблица 3 – Сравнительная таблица времени выполнения

Метрика	До оптимизации	После оптимизации	Вывод
План (Оператор)	Seq Scan	Bitmap Index Scan	Выбор отличается
Execution Time	6.797 ms	0.755 ms	Запрос **ускорился** примерно в 9 раз

2.4 ЗАДАНИЕ 2. ДЕМОНСТРАЦИЯ АТОМАРНОСТИ. УСПЕШНЫЙ COMMIT

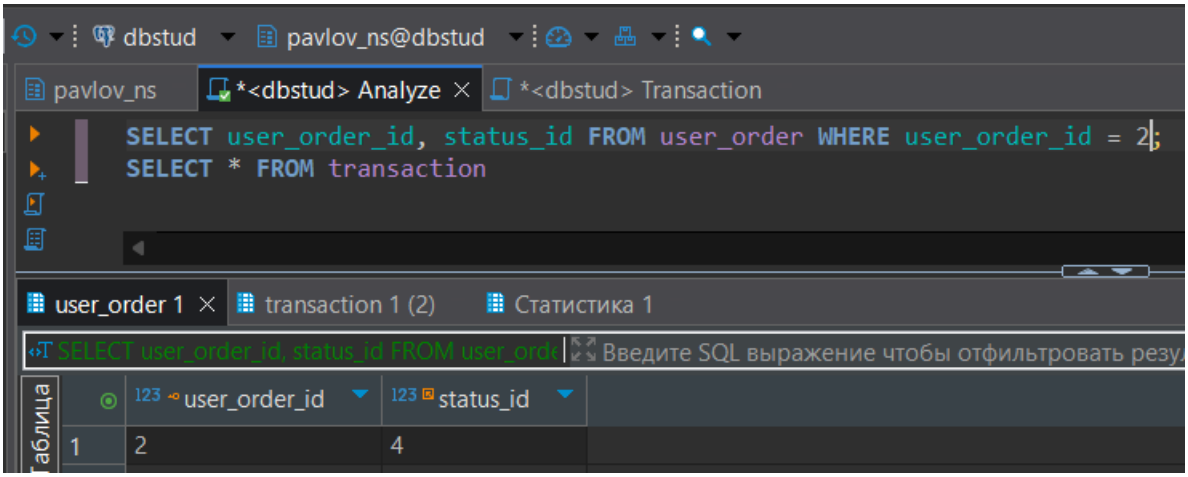


Рисунок 16 – Проверка исходного статуса заказа

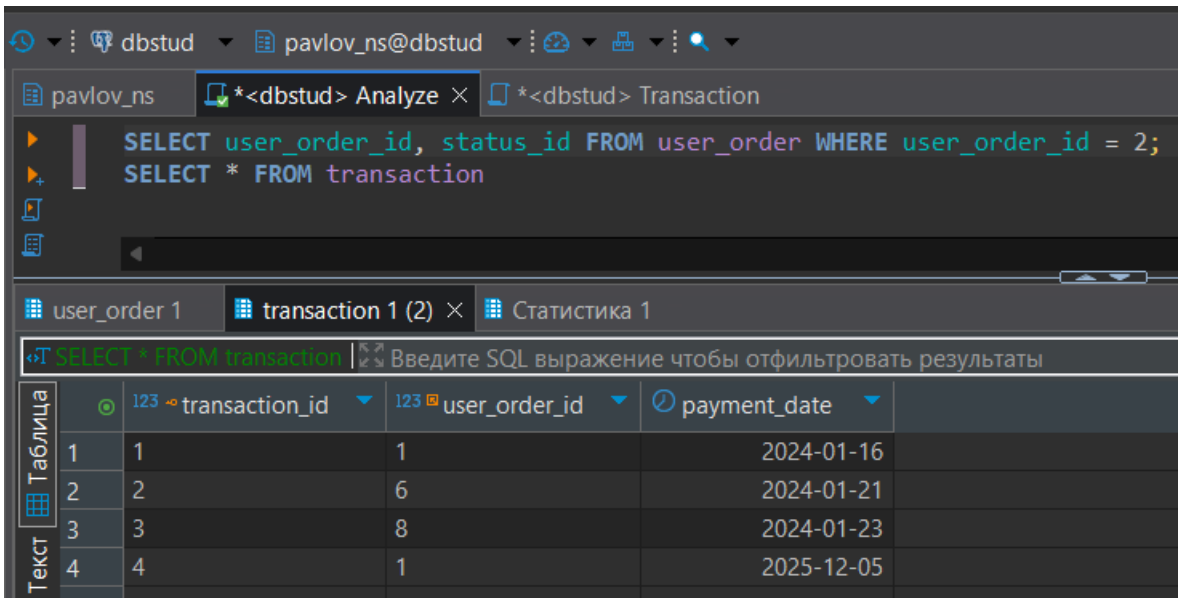


Рисунок 17 – Проверка исходного состояния таблицы transaction

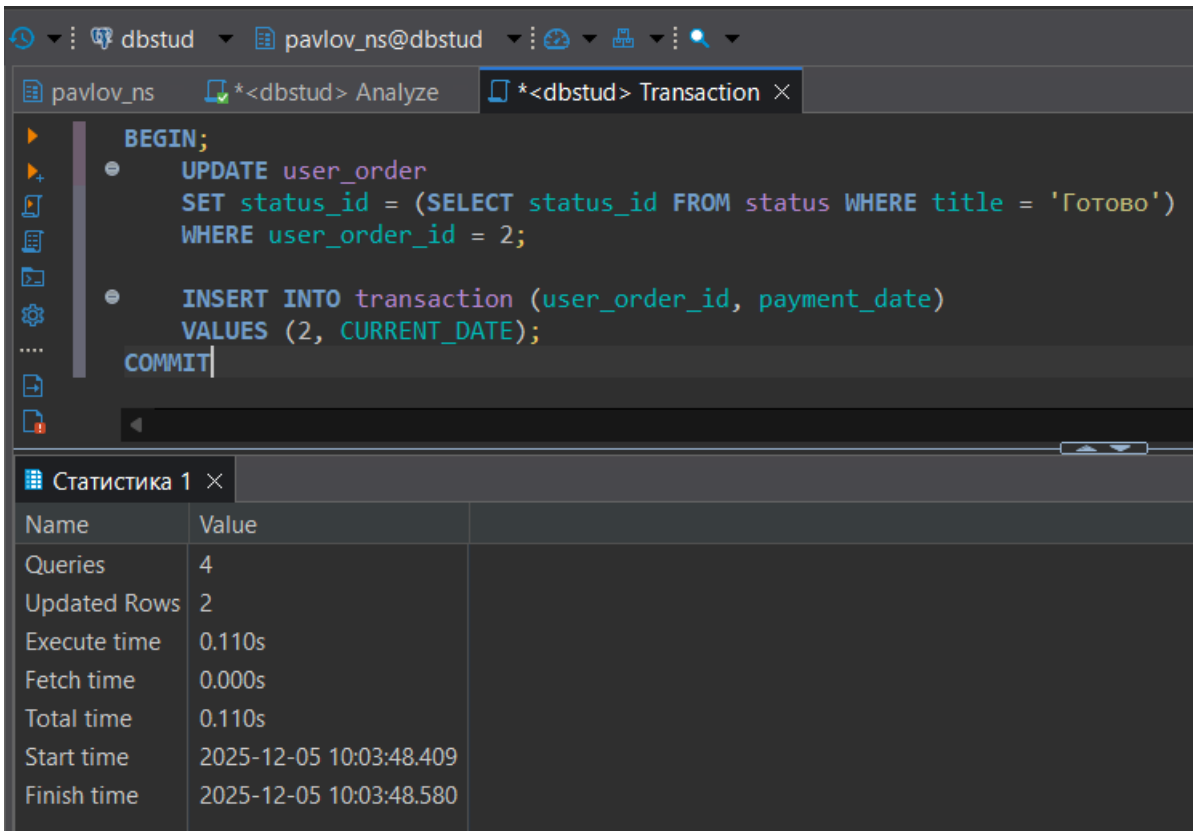


Рисунок 18 – Выполнение атомарной операции обновления данных

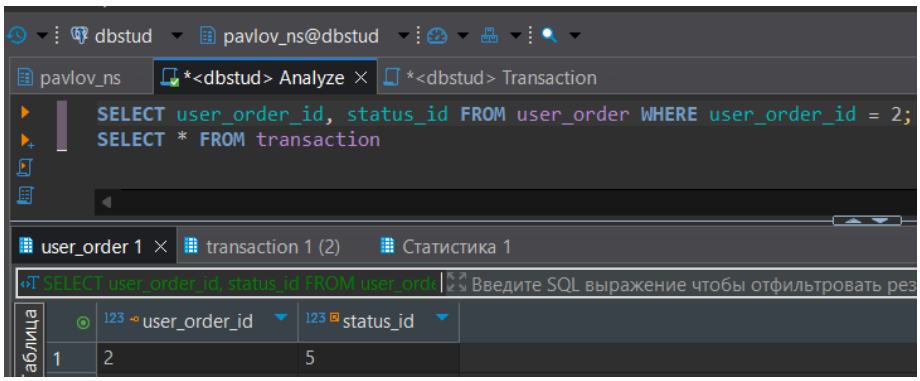


Рисунок 19 – Проверка обновления статуса заказа

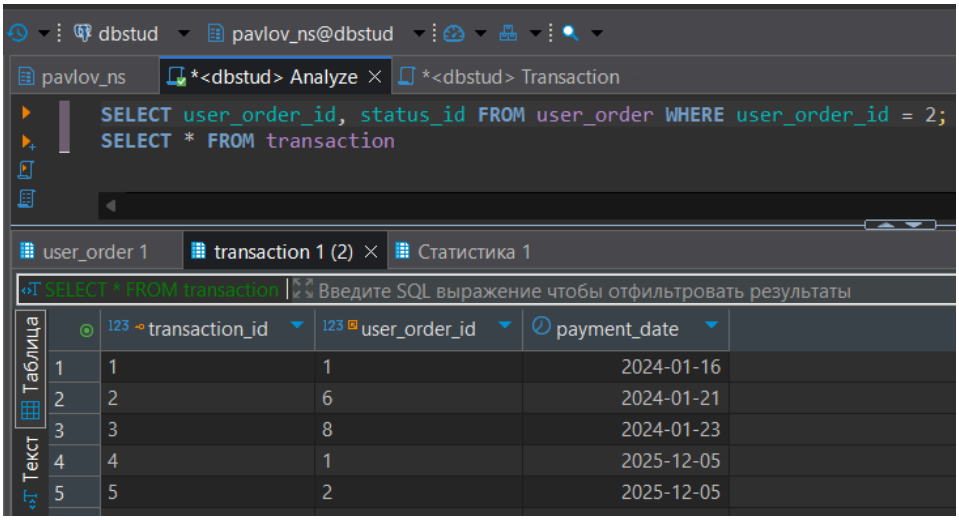


Рисунок 20 – Проверка обновления таблицы transaction

2.5 ЗАДАНИЕ 3. ДЕМОНСТРАЦИЯ АТОМАРНОСТИ. АВАРИЙНЫЙ ROLLBACK

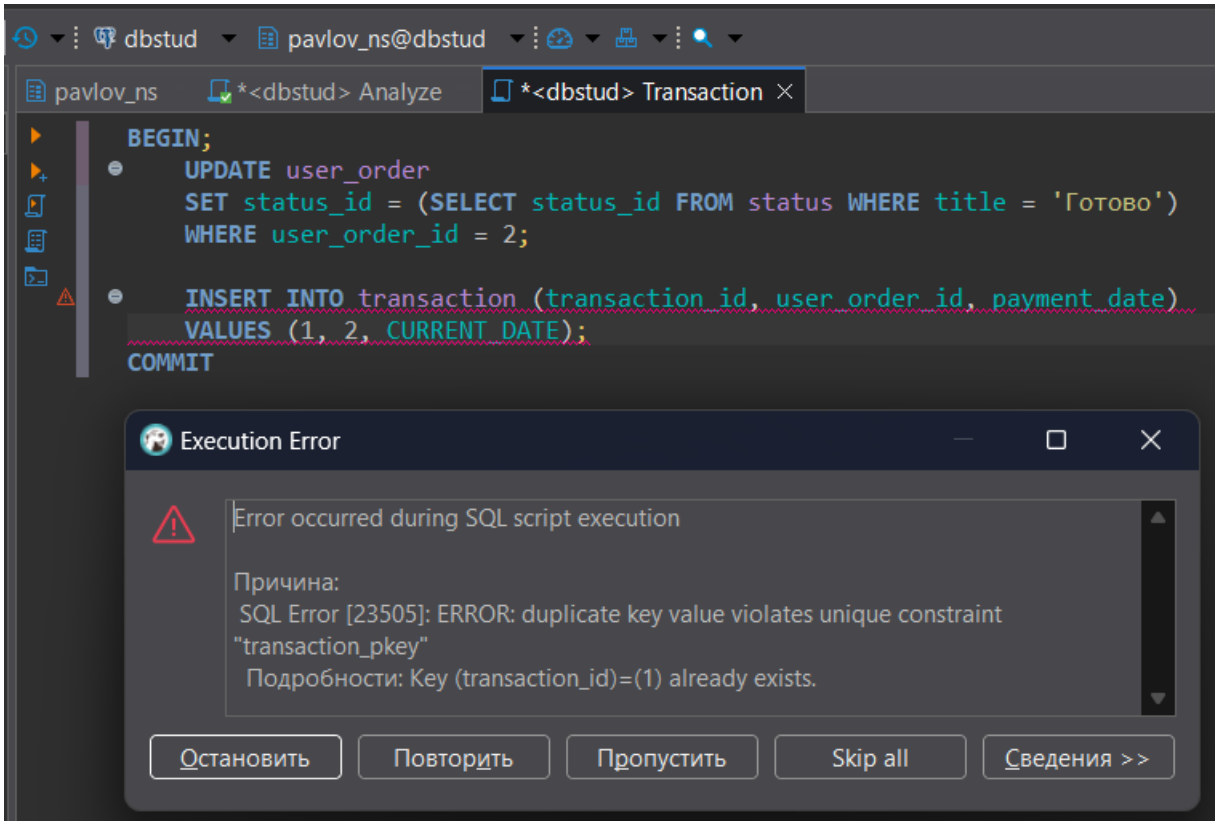


Рисунок 21 – Выполнение запроса с ошибкой

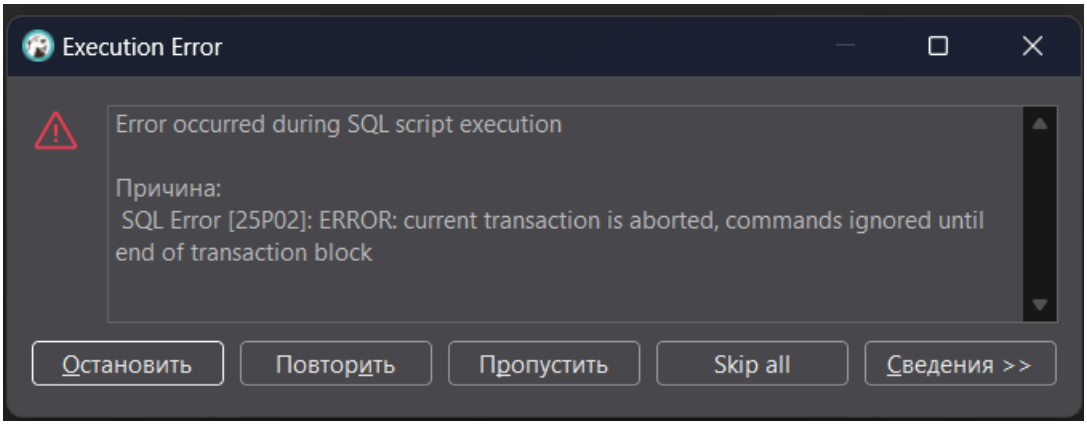


Рисунок 22 – Ошибка при попытке выполнить любой запрос

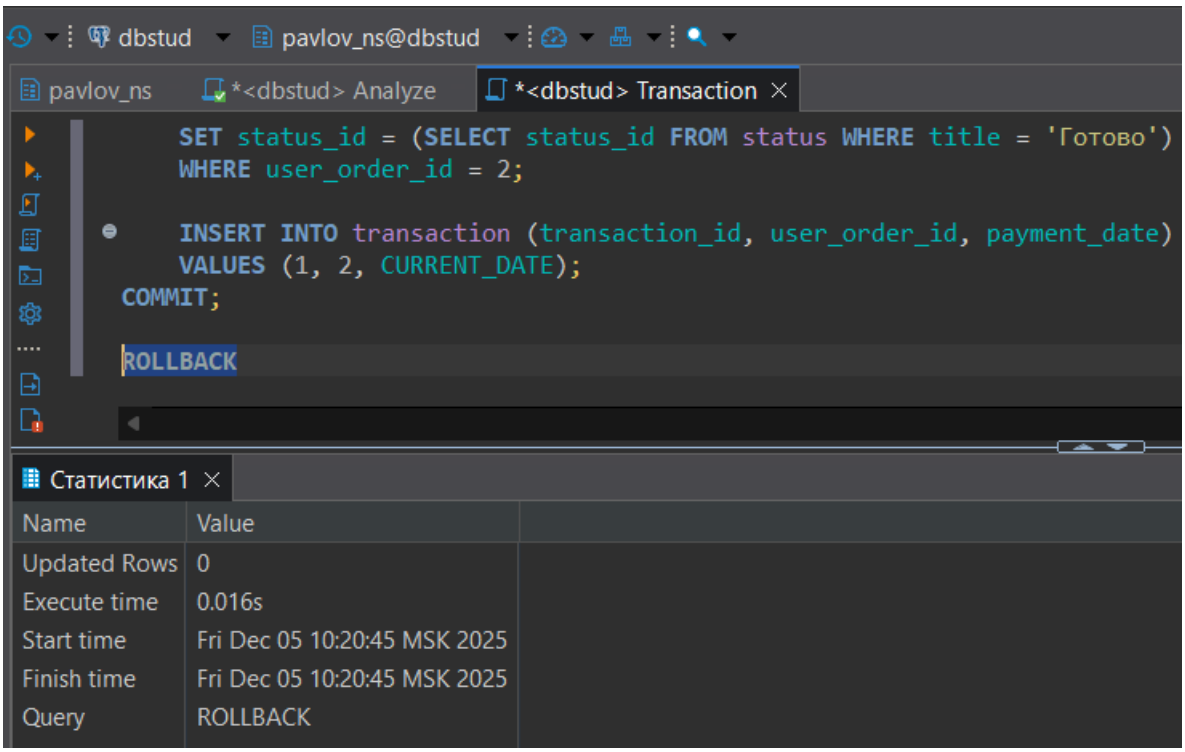


Рисунок 23 – Выполнение ROLLBACK

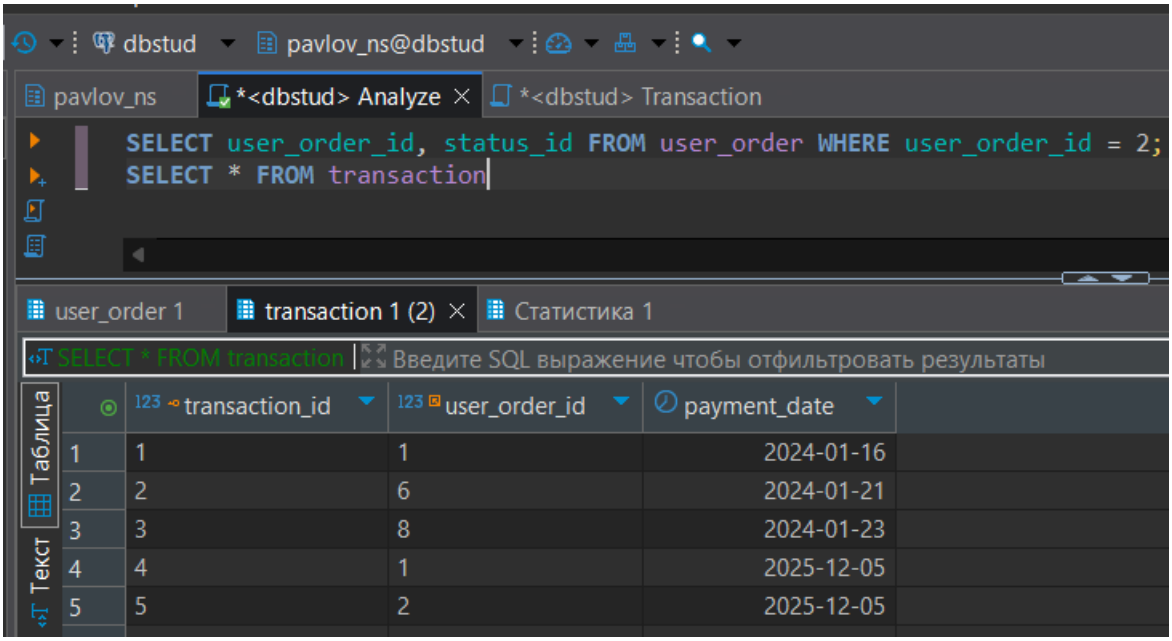


Рисунок 24 – Проверка состояния после ROLLBACK

2.6 ЗАДАНИЕ 4. МОДЕЛИРОВАНИЕ АНОМАЛИИ «НЕПОВТОРЯЕМОЕ ЧТЕНИЕ»

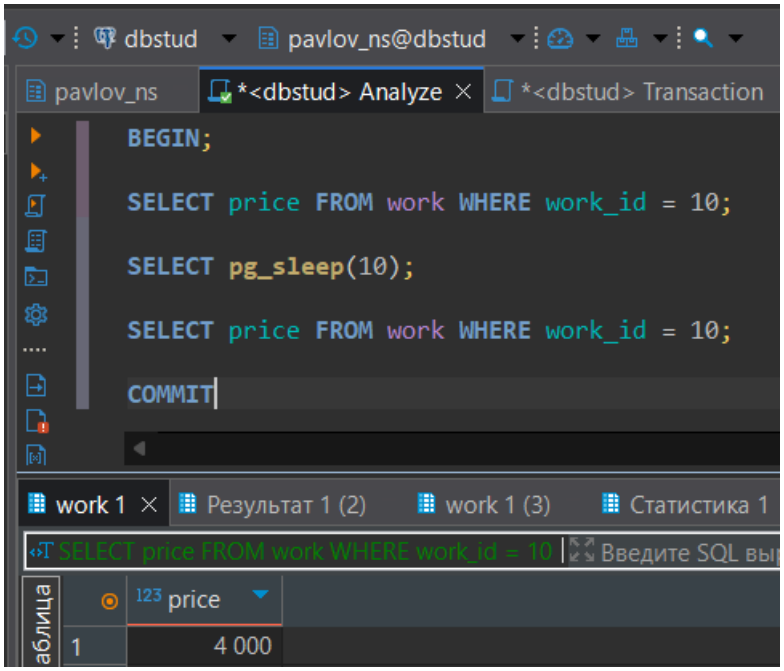


Рисунок 25 – Вывод при первом чтении

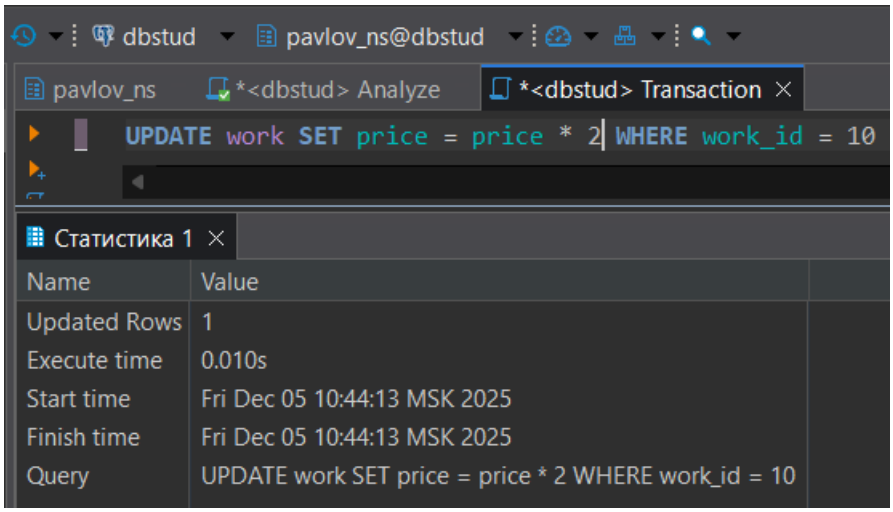


Рисунок 26 – Обновление данных во время ожидания

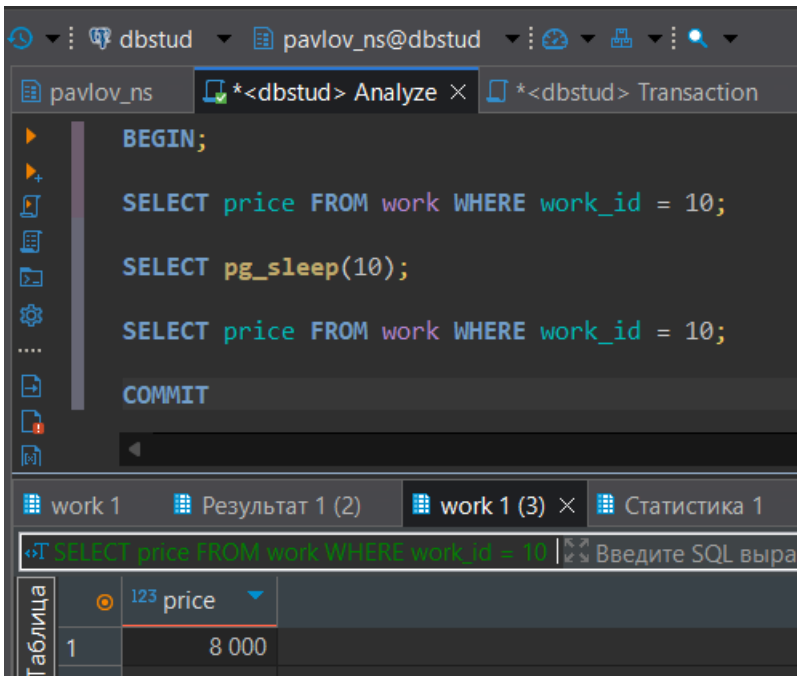


Рисунок 27 – Вывод при втором чтении

2.7 ЗАДАНИЕ 5. УСТРАНЕНИЕ АНОМАЛИИ «НЕПОВТОРЯЕМОЕ ЧТЕНИЕ»

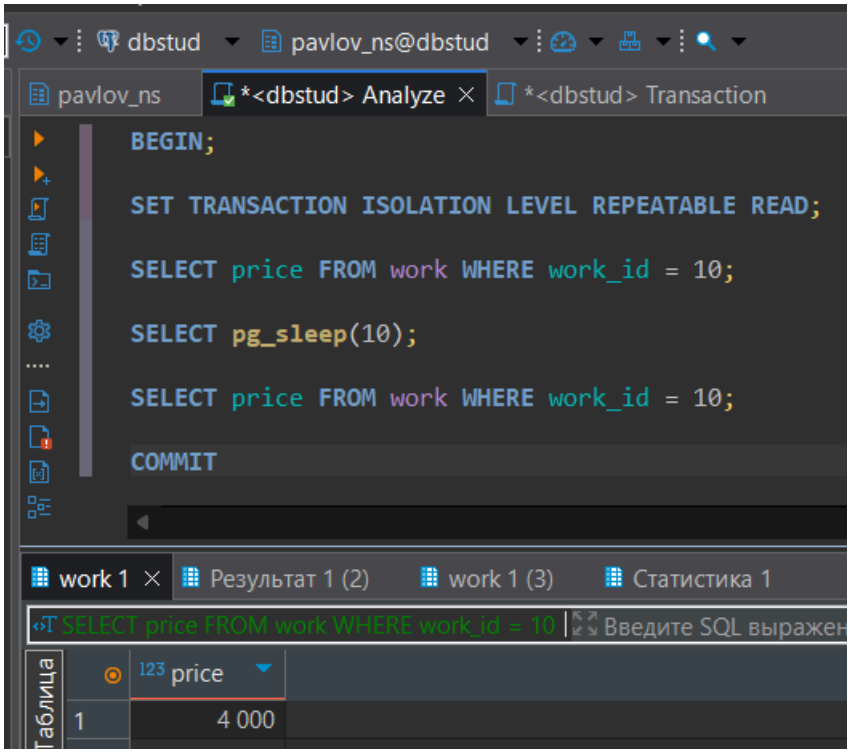


Рисунок 28 – Вывод при первом чтении

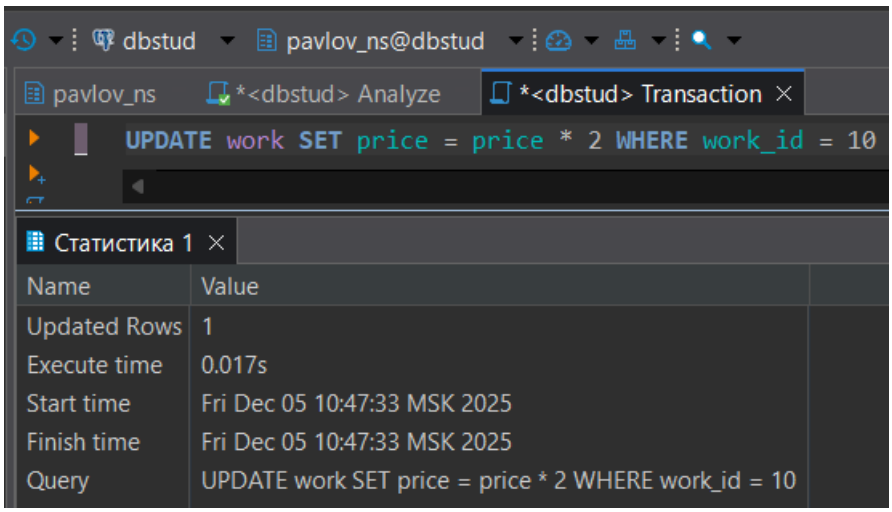


Рисунок 29 – Обновление данных во время ожидания

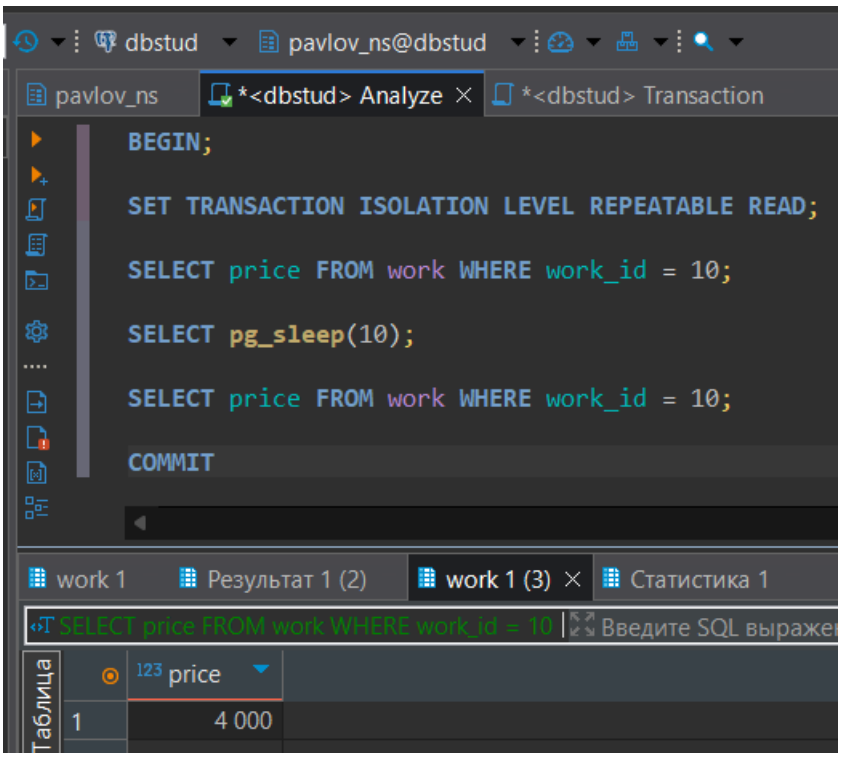


Рисунок 30 – Вывод при втором чтении