|  |  |  |
| --- | --- | --- |
|  |  |  |
| МИНОБРНАУКИ РОССИИ | | |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** | | |

Институт Информационных технологий (ИТ)

Кафедра Инструментального и прикладного программного обеспечения

(ИиППО)

**ОТЧЕТ О ПРАКТИЧЕСКОЙ РАБОТЕ №2**

**по дисциплине**

«Технологии обработки транзакций клиент-серверных приложений»

Выполнил студент группы ИКБО-20-21 Хитров Н.С.

Принял Маличенко С.В.

Москва 2024

**ЗАДАНИЕ НА ПРАКТИЧЕСКУЮ РАБОТУ**

**Работа с Rollback**

Начните транзакцию (командой BEGIN) и создайте новый заказ в таблице sales\_order с сегодняшней датой. Добавьте два предмета в таблицу item, связанных с созданным заказом.

Представьте, что пользователь не подтвердил заказ и все введённые данные необходимо отменить. Выполните отмену транзакции и проверьте, что никакой добавленной вами информации действительно не осталось.

Теперь представьте сценарий, в котором нужно отменить не все данные, а только последний из добавленных предметов. Для этого повторите все действия из предыдущего упражнения, но перед добавлением каждого предмета создавайте точку сохранения (с одним и тем же именем). После ввода второго предмета выполните откат к точке сохранения. Проверьте, что заказ и первый предмет остались.

В рамках той же транзакции добавьте еще один предмет и зафиксируйте транзакцию. Обратите внимание на то, что после этой операции отменить внесённые транзакцией изменения будет уже невозможно.

**Уровень изоляции Read Committed**

Перед началом выполнения задания проверьте, что в таблице sales\_order нет заказов на сумму total 1 000 рублей.

В первом сеансе начните транзакцию (командой BEGIN). Выполните обновление таблицы sales\_order: увеличьте total в два раза в тех строках, где сумма равна 1 000 рублей.

Во втором сеансе (откройте новое окно psql). Начните транзакцию (командой BEGIN). Вставьте в таблицу sales\_order новый заказ на 1 000 рублей и зафиксируйте транзакцию.

В первом сеансе повторите обновление таблицы sales\_order и зафиксируйте транзакцию.

Осталась ли сумма добавленного заказа равной 1 000 рублей? Почему это не так?

**Уровень изоляции Repeatable Read**

Повторите предыдущее упражнение, но начните транзакцию в первом сеансе с уровнем изоляции транзакций Repeatable Read. Объясните различие полученных результатов.

Выполните указанные действия в двух сеансах:

В первом сеансе начните новую транзакцию с уровнем изоляции Repeatable Read. Вычислите количество заказов с суммой 20 000 рублей.

Во втором сеансе начните новую транзакцию с уровнем изоляции Repeatable Read. Вычислите количество заказов с суммой 30 000 рублей.

В первом сеансе добавьте новый заказ на 30 000 рублей и снова вычислите количество заказов с суммой 20 000 рублей.

Во втором сеансе добавьте новый заказ на 20 000 рублей и снова вычислите количество заказов с суммой 30 000 рублей.

Зафиксируйте транзакции в обоих сеансах.

Соответствует ли результат ожиданиями? Можно ли сериализовать эти транзакции (иными словами, можно ли представить такой порядок последовательного выполнения этих транзакций, при котором результат совпадает с тем, что получился при параллельном выполнении)?

**РЕФЕРАТ**

Отчет 16 страниц, 9 рисунков, 5 источников.

POSTGRESQL, УРОВЕНЬ ИЗОЛЯЦИИ, ТРАНЗАКЦИИ, НЕПОВТОРЯЕМОЕ ЧТЕНИЕ, SQL

Объектом разработки является база данных.

Цель работы – разработка и заполнение базы данных.

В процессе работы проводилось практическое использование базы данных в виде создания и заполнения различных таблиц.

Результатом является подготовленное рабочее место с установленным PostgreSQL и заполненная база данных.

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 7](#_Toc159618609)

[1. ВЫПОЛНЕНИЕ ПРАКТИЧЕСКОЙ РАБОТЫ 8](#_Toc159618610)

[1.1 Базовая работа с транзакциями 8](#_Toc159618611)

[1.2 Уровень изоляции Read Committed 11](#_Toc159618612)

[1.3 Уровень изоляции Repeatable Read 12](#_Toc159618613)

[ЗАКЛЮЧЕНИЕ 15](#_Toc159618614)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 16](#_Toc159618615)

**ПЕРЕЧЕНЬ СОКРАЩЕНИЙ И ОБОЗНАЧЕНИЙ**

БД – база данных;

СУБД – система управления базами данных.

**ВВЕДЕНИЕ**

В современной сфере баз данных, PostgreSQL занимает выдающуюся позицию, предоставляя мощный инструмент для управления информацией. Эта система управления базами данных предлагает механизм транзакций, который играет ключевую роль в обеспечении последовательности и консистентности изменений в базе данных.

Транзакции в PostgreSQL выполняют множество задач, включая поддержку атомарности, согласованности, изолированности и долговечности (ACID). Эти принципы являются фундаментальными для обеспечения надёжности и предсказуемости работы с данными. В данном контексте, изучение и понимание принципов транзакций в PostgreSQL становится необходимым для разработчиков и администраторов баз данных, стремящихся обеспечить эффективное управление информацией в сложных и динамичных проектах.

Целью данной работы является практическое исследование аспектов работы с транзакциями в базах данных на примере системы управления PostgreSQL. Работа сфокусирована на сценариях отмены транзакций, рассматривается уровни изоляции и их влияние на консистентность данных. Работа предоставляет конкретные примеры, демонстрируя важность правильного управления транзакциями в реальных сценариях.

1. **ВЫПОЛНЕНИЕ ПРАКТИЧЕСКОЙ РАБОТЫ**

## Базовая работа с транзакциями

На рисунке 1 изображено открытие транзакции при помощи команды «BEGIN». Далее, выполнение внутри транзакции вставки строк в таблицы «sales\_order» (1 строка) и «item» (2 строки). После чего транзакция была отменена командой «ROLLBACK» и результат выполнения команд был выведен в консоль.

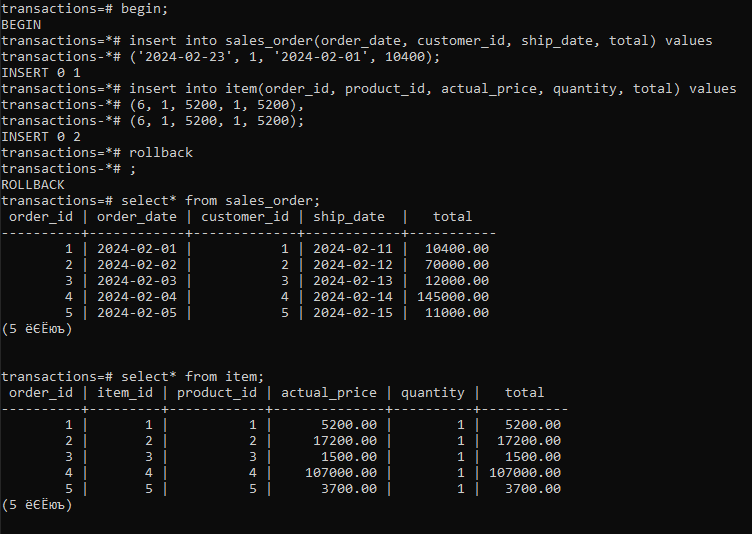


Рисунок 1 – Запуск транзакции и отмена ее действий

В рамках выполнения следующей задачи была начата новая транзакция, в которой перед созданием каждой строки таблицы «item» были созданы точки сохранения с одинаковыми названиями при помощи команды «SAVEPOINT». После был выполнен откат до точки сохранения командой «ROLLBACK TO». Далее была успешно выполнена проверка совпадения результата с ожидаемым. (Рисунок 2)

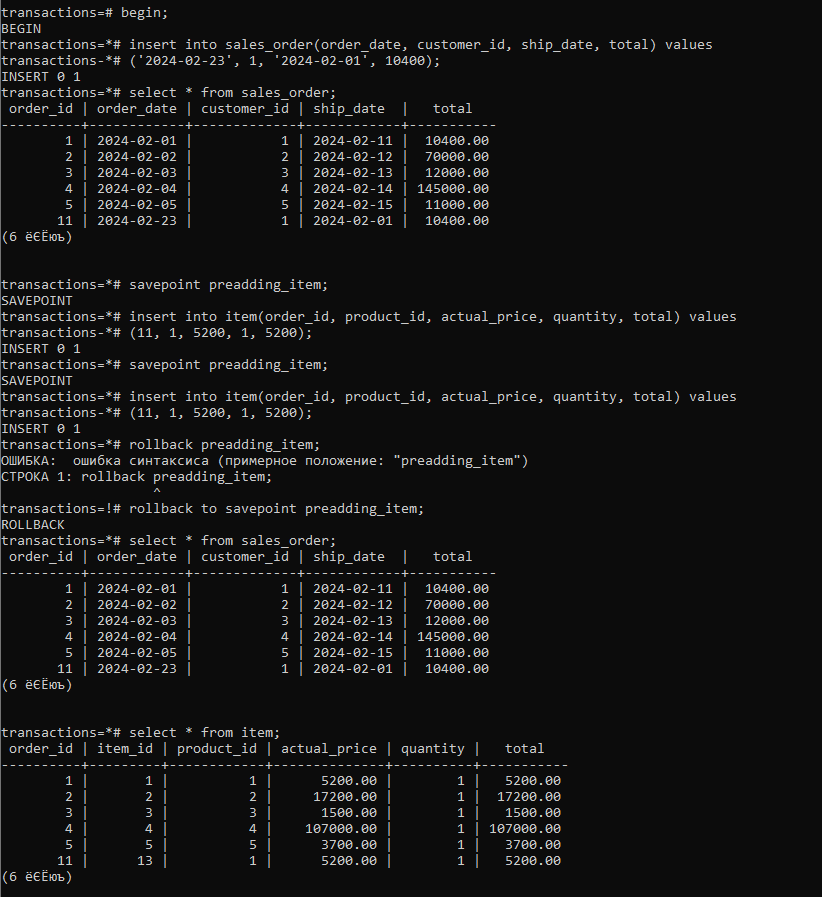


Рисунок 2 - Запуск транзакции и окат ее действий до последней точки сохранения

Согласно следующей части задания была добавлена еще одна строка таблицы «item» и изменения были зафиксированы, как представлено на рисунке 3. Также на рисунке показана проверка результата.

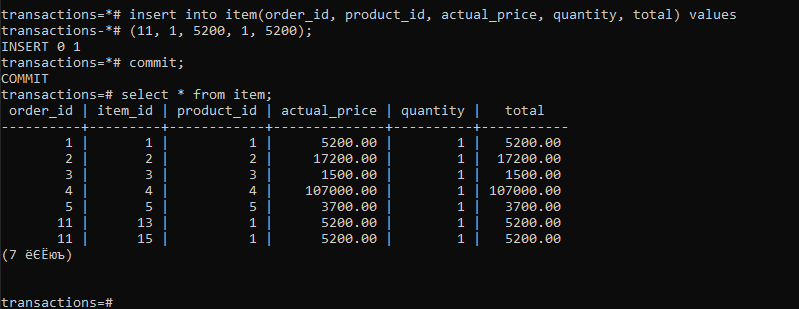


Рисунок 3 - Добавление данных в таблицу item и фиксация транзакции

## Уровень изоляции Read Committed

После проверки таблицы «sales\_order» на наличие заказов на 1000 рублей в первом сеансе была открыта транзакция в которой было выполнено увеличение стоимости заказов вдвое во всех строках таблицы, содержащих значение 1000 в столбце «total» (Рисунок 4). Далее во втором сеансе была запущена и зафиксирована транзакция, в которой была выполнена операция вставки заказа на сумму 1000 рублей (Рисунок 5). В первом сеансе был осуществлен повтор обновления таблицы, таким образом, что все строки со стоимостью заказа равной 1000 увеличивают стоимость вдвое, после чего транзакция была зафиксирована (Рисунок 4)

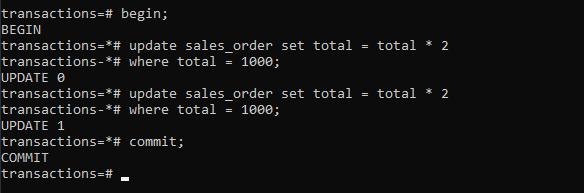


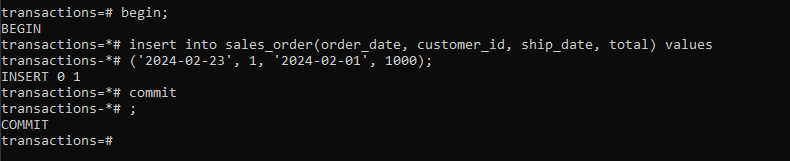
Рисунок 4 – Работа с транзакцией в первом сеансе

Рисунок 5 – Работа с транзакцией во втором сеансе

В результате выполнения транзакций сумма добавленного товара стала равной 2000 рублей, так как транзакции были запущены в стандартном режиме изоляции «Read committed», который позволяет выполнять чтение изменений, зафиксированных в транзакциях других сеансов.

## Уровень изоляции Repeatable Read

Предыдущее задание было повторено с запуском транзакции первого сеанса с уровнем изоляции «Repeatable read» (Рисунки 6-7). Были получены отличные от предыдущего задания результаты, так как repeatable read уровень изоляции не выполнять чтение изменений, параллельно зафиксированных в транзакциях других сеансов.

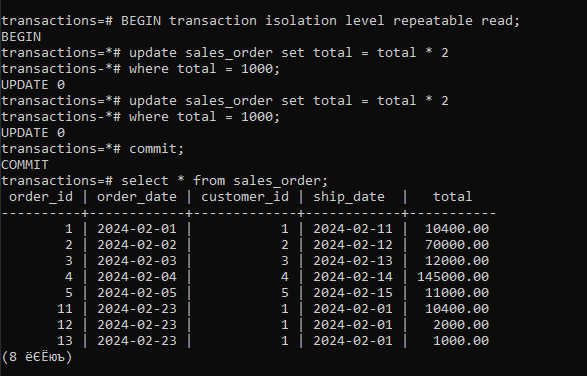


Рисунок 6 – Начало транзакции с уровнем изоляции Repeatable Read и удвоение поля total в некоторых строках

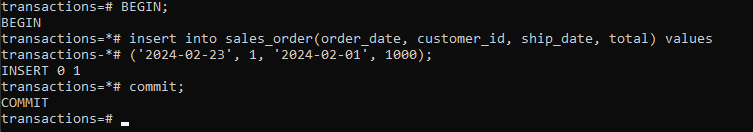


Рисунок 7 – Начало транзакции в сессии 2 и добавление новой записи в sales\_order с фиксацией изменений

Далее на рисунках 8-9 представлено выполнение следующих пунктов работы:

* В первом сеансе была начата новая транзакция с уровнем изоляции Repeatable Read. Было вычислено количество заказов с суммой 20 000 рублей (0).
* Во втором сеансе была начата новая транзакция с уровнем изоляции Repeatable Read. Было вычислено количество заказов с суммой 30 000 рублей (0).
* В первом сеансе был добавлен новый заказ на 30 000 рублей и снова вычислено количество заказов с суммой 20 000 рублей (результат не изменился).
* В первом сеансе был добавлен новый заказ на 20 000 рублей и снова вычислено количество заказов с суммой 30 000 рублей (результат не изменился).
* Были зафиксированы транзакции в обоих сеансах.

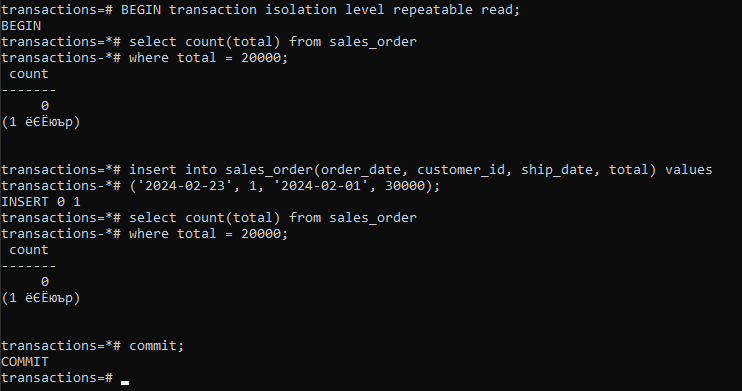


Рисунок 8 – Работа с Repeatable Read транзакцией в первом сеансе

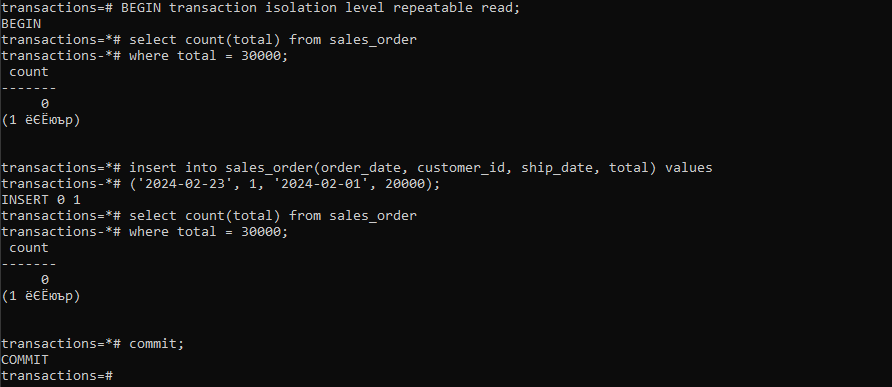


Рисунок 9 – Работа с Repeatable Read транзакцией во втором сеансе

Результат соответствует ожиданиям. Данные транзакции невозможно сериализовать, так чтобы порядок последовательного и параллельного выполнения вызывали одинаковый результат, так как сериализация приведет к тому, что первый сеанс будет видеть обновления от транзакций второго сеанса и наоборот.

# ЗАКЛЮЧЕНИЕ

В ходе выполнения работы были изучены основные уровни изоляции транзакций, проблемы, которые они решают, а также и их отличия.

На практике было освоено использование уровней изоляции транзакций: Read Commited, Repeatable Read; использование команд: «BEGIN», «SAVEPOINT», «ROLLBACK», «COMMIT» для запуска транзакций, в том числе с заданием уровня изоляции, для создания точек сохранения внутри транзакции, для отката транзакции, в том числе с указанием крайней точки сохранения, к которой выполняется откат, для фиксации транзакций.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. PostgreSQL: Официальная документация [Электронный ресурс] – URL: https://www.postgresql.org/docs/ (дата обращения: 14.02.2024).
2. PostgreSQL Tutorial for Beginners [Электронный ресурс] – URL: https://www.tutorialspoint.com/postgresql/index.htm (дата обращения: 14.02.2024).
3. PostgreSQL: Википедия [Электронный ресурс] – URL: https://en.wikipedia.org/wiki/PostgreSQL (дата обращения: 14.02.2024).
4. Лузанов П.В. и др. Postgres. Первое знакомство. [Электронный ресурс] – URL: https://postgrespro.ru/education/books/introbook (дата обращения: 22.02.2024)
5. Новиков Б. А. Лекции Основы технологий баз данных. [Электронный ресурс] – URL: https://postgrespro.ru/education/university/dbtech (дата обращения: 22.02.2024)