

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«МИРЭА – Российский технологический университет»**

**РТУ МИРЭА**

|  |
| --- |
|  |

Институт информационных технологий

Кафедра инструментального и прикладного программного обеспечения (ИППО)

**ОТЧЕТ**

**ПО ПРАКТИЧЕСКОЙ РАБОТЕ №7**

**по дисциплине**

«Технологии обработки транзакций клиент-серверных приложений»

Выполнил студент группы ИКБО-16-21 Белослудцев Е.Д.

Принял Маличенко С. В.

Практическая работа выполнена «\_\_» \_\_\_\_\_\_\_ 2024 г.

«Зачтено» «\_\_» \_\_\_\_\_\_\_ 2024 г.

Москва 2024

**ЗАДАНИЕ НА ПРАКТИЧЕСКУЮ РАБОТУ**

Необходимо выполнить следующие шаги:

1. Узнать какие блокировки на уровне изоляции Read Committed удерживает транзакция, прочитавшая одну строку таблицы по первичному ключу.
2. Повторить предыдущий пункт для уровня изоляции Serializable.
3. Настроить сервер так, чтобы в журнал сообщений сбрасывалась информация о блокировках, удерживаемых более 100 миллисекунд. Воспроизвести ситуацию, при которой в журнале появятся такие сообщения.

**РЕФЕРАТ**

Белослудцев Е.Д., Практическая работа направления подготовки «Программная инженерия» на тему «Транзакции. Блокировки. Блокировки объектов»: М. 2024 г., МИРЭА – Российский технологический университет (РТУ МИРЭА), Институт информационных технологий (ИИТ), кафедра инструментального и прикладного программного обеспечения (ИиППО) – 15 стр., 12 рис., 9 источн.

Ключевые слова: POSTGRESQL, БАЗА ДАННЫХ, ТАБЛИЦА, SQL, ТРАНЗАКЦИЯ, БЛОКИРОВКА, SERIALIZABLE, READ COMMITTED.

Целью работы является исследование различных типов блокировок и механизмов их действия с целью оптимизации использования ресурсов.

Belosludtsev E.D., Practical work of the direction of training “Software Engineering” on the theme “Transactions. Locks. Object locks”: M. 2024, MIREA – Russian Technological University (RTU MIREA), Institute of Information of Instrumental and Applied Software (IAPS) – 15 p., 12 fig., 9 sources.

Keywords: POSTGRESQL, DATABASE, TABLE, SQL, TRANSACTION, LOCK, SERIALIZABLE, READ COMMITTED.

The aim of the work is to study different types of locks and their mechanisms of action in order to optimize resource utilization.

**ПЕРЕЧЕНЬ СОКРАЩЕНИЙ**

|  |  |  |
| --- | --- | --- |
| SQL | — | Structured Query Language |
| БД | — | База данных |
| СУБД | — | Система управления базой данных |
|  |  |  |

**ПЕРЕЧЕНЬ ТЕРМИНОВ И ОПРЕДЕЛЕНИЙ**

1. **База данных** – совокупность данных, хранимых в соответствии со схемой данных, манипулирование которыми выполняют в соответствии с правилами средств моделирования данных.
2. **Таблица** – совокупность связанных данных, хранящихся в структурированном виде в базе данных.
3. **SQL-скрипт** – сохраненная в текстовом формате SQL инструкция, с помощью которой создаются объекты базы данных.
4. **Транзакция** – последовательность операторов языка SQL, которая рассматривается как некоторое неделимое действие над базой данных, осмысленное с точки зрения пользователя.
5. **Блокировка** – метод предотвращения одновременного доступа к данным в БД, позволяющий обеспечить отсутствие противоречивых результатов.

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 7](#_Toc162392321)

[ХОД РАБОТЫ 8](#_Toc162392322)

[ЗАКЛЮЧЕНИЕ 13](#_Toc162392323)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 14](#_Toc162392324)

# ВВЕДЕНИЕ

В реляционных базах данных транзакции играют важную роль в поддержании интегритета и согласованности данных. Транзакция представляет собой логическую единицу работы, состоящую из одной или нескольких операций, которые должны быть либо полностью завершены, либо полностью отменены. Это обеспечивает постоянное соблюдение согласованного состояния данных, даже в случае сбоев или ошибок.

Для поддержания целостности транзакций и предотвращения конфликтов при одновременном доступе к данным из разных транзакций используются механизмы блокировок. Блокировки позволяют регулировать доступ к данным, обеспечивая, что только одна транзакция может изменять определенные данные в конкретный момент времени.

Существует два основных вида блокировок:

Блокировки ресурсов: они применяются к целым ресурсам базы данных, таким как таблицы или страницы данных. Они предотвращают конфликты доступа, блокируя весь ресурс для операций чтения или записи.

Блокировки записей: применяются на уровне отдельных записей данных. Они более тонко управляют параллелизмом, разрешая одновременный доступ к разным записям одной таблицы из разных транзакций.

Корректное управление транзакциями и блокировками имеет критическое значение для обеспечения целостности данных, предотвращения потери информации и обеспечения согласованного состояния базы данных при параллельном доступе. Кроме того, это помогает оптимизировать производительность базы данных и обеспечить ее надежность.

# ХОД РАБОТЫ

Чтобы узнать какие блокировки на уровне изоляции Read Committed удерживает транзакция, прочитавшая одну строку таблицы по первичному ключу, нужно получить идентификатор процесса после, начать транзакцию и выполнить чтение определенной записи – рисунок 1.

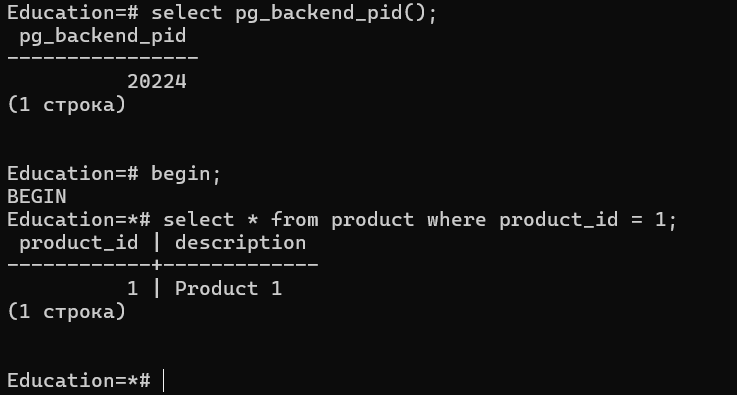


Рисунок 1 – Начало транзакции (Read Committed) и чтение одной строки

Следующим шагом мы определяем какие блокировки удерживает транзакция при чтении непосредственно в самой транзакции – рисунок 2.

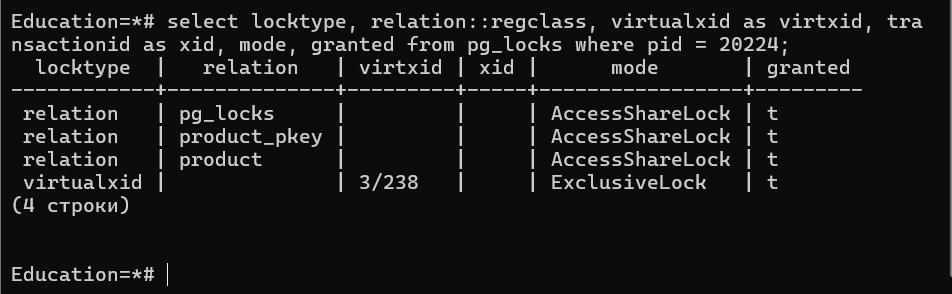


Рисунок 2 – Вывод блокировок в самой транзакции (Read Committed)

А также сделаем вывод блокировок вне транзакции– рисунок 3.

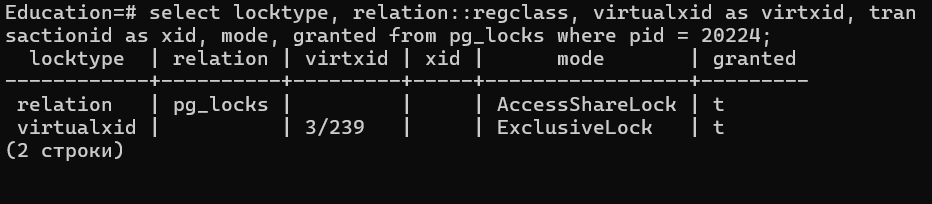


Рисунок 3 – Вывод блокировок читающей транзакции (Read Committed)

В общей таблице блокировок (рис. 3) отображаются следующие блокировки:

* блокировка на таблицу pg\_locks в режиме AccessShareLock
* исключительная блокировка собственного номера виртуальной транзакции.

В таблице блокировок в самой транзакции (рис. 2) к ним добавится следующие блокировки:

* блокировка индекса product\_pkey, созданного для первичного ключа в режиме AccessShareLock;
* блокировка таблицы product в режиме AccessShareLock.

Предыдущие шаги также выполнены для уровня изоляции Serializable (рис. 4-6).

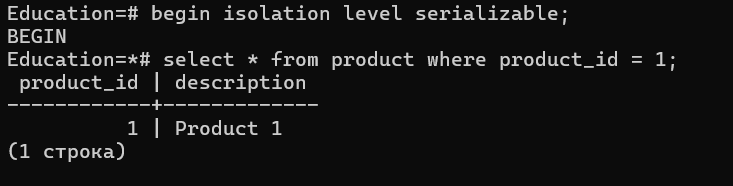


Рисунок 4 – Начало транзакции (Serializable) и чтение одной строки

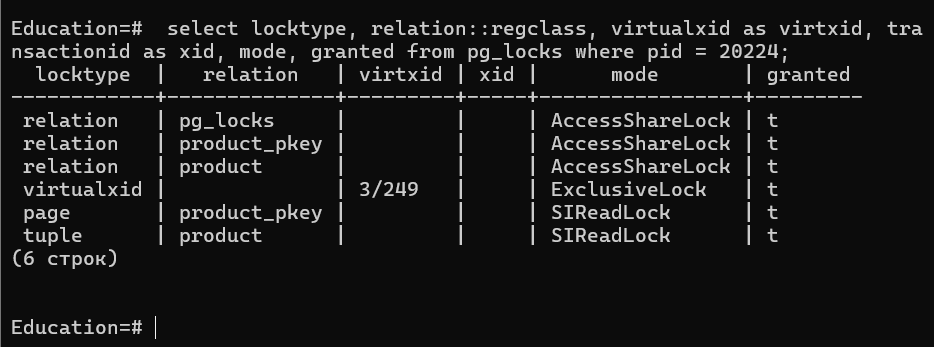


Рисунок 5 – Вывод блокировок читающей транзакции (Serializable)

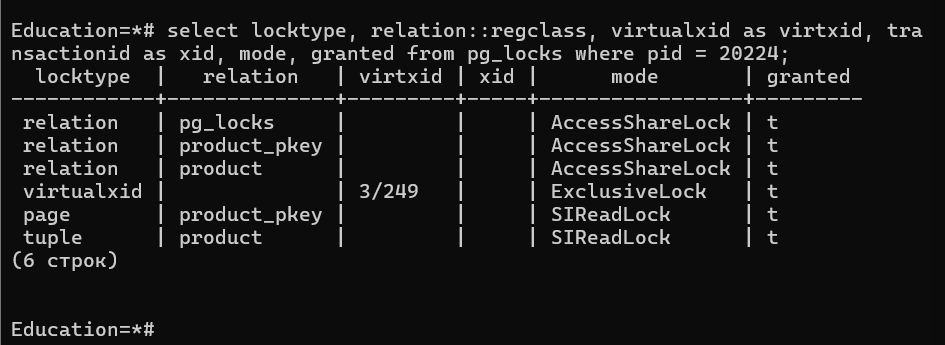
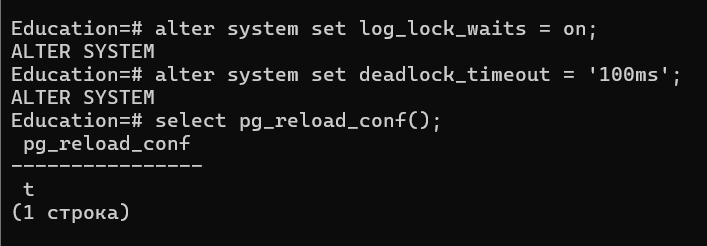


Рисунок 6 – Вывод блокировок в самой транзакции (Serializable)

К предыдущим блокировка добавились:

* предикатная блокировка страницы индекса;
* предикатная блокировка прочитанной версии строки

Далее, сервер настроен так, чтобы в журнал сообщений сбрасывалась информация о блокировках, удерживаемых более 100 мс и воспроизведена ситуация, при которой в журнале появятся такие сообщения – рисунок 7.

  
Рисунок 7 – Настройка сервера

Следующим шагом будет воспроизведение начало транзакции и внесение некоторых изменений в структуру БД – рисунок 8.

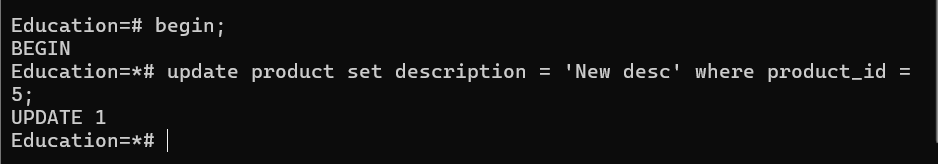


Рисунок 8 – Начало первой транзакции и внесение изменений

Повторим предыдущий шаг, только для второй транзакции – рисунок 9.

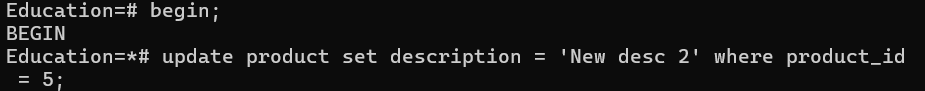


Рисунок 9 – Начало второй транзакции и попытка внесения изменений

Приостанавливаем работу текущего сеанса с помощью команды pg\_sleep и выполняет фиксацию изменений – рисунок 10.

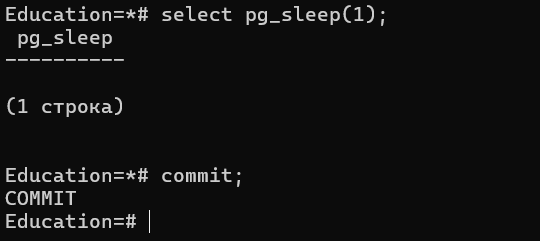


Рисунок 10 – Выполнение задержки в первой транзакции и ее завершения

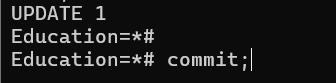


Рисунок 11 – Фиксация второй транзакции

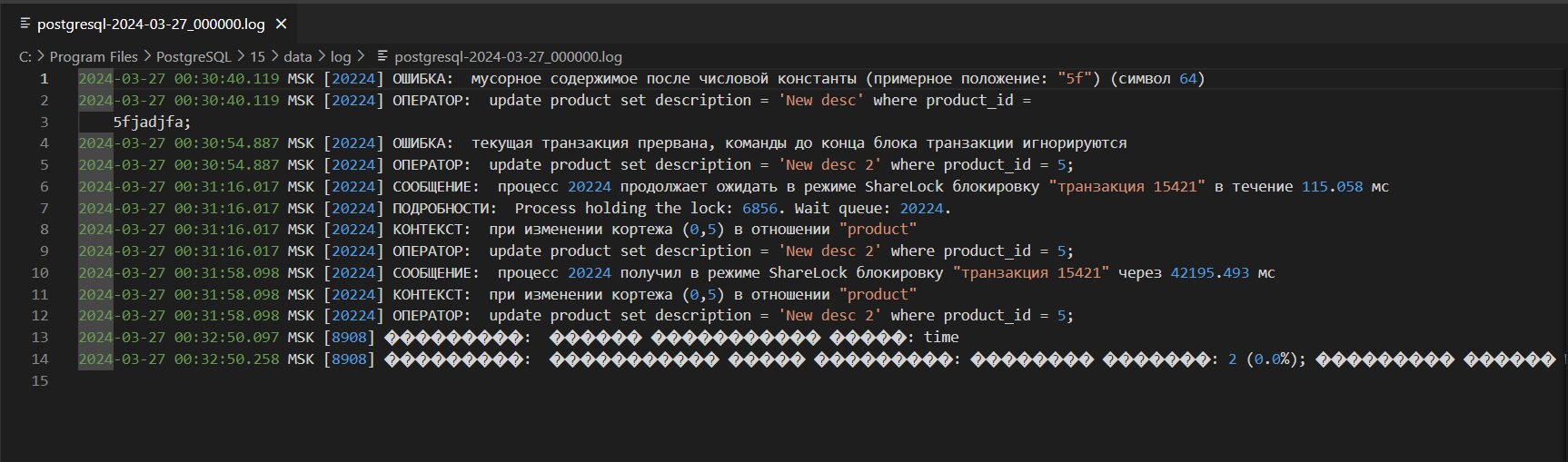


Рисунок 12 – Логи журнала сообщений

# ЗАКЛЮЧЕНИЕ

В ходе исследования были рассмотрены различные типы блокировок и механизмы их действия с целью оптимизации использования ресурсов базы данных. На уровне изоляции Read Committed выяснено, что транзакция, прочитавшая одну строку по первичному ключу, удерживает блокировку только на эту строку на время чтения, предотвращая грязное чтение. Это позволяет другим транзакциям параллельно читать и модифицировать другие строки той же таблицы.

При уровне изоляции Serializable было выявлено, что транзакция при чтении строки держит блокировку на более высоком уровне, предположительно, на таблицу, гарантируя, таким образом, сериализуемую изоляцию. Это означает, что никакие другие транзакции не могут параллельно модифицировать строки этой таблицы. Однако, уровень Serializable может вызвать проблемы с производительностью при высокой конкуренции из-за увеличенной конкуренции блокировок.

В процессе выполнения практических задач были продемонстрированы различия между режимами изоляции и выявлена важность понимания различных видов блокировок для обеспечения целостности данных при параллельном доступе. Была подчеркнута также польза логгирования ситуаций с блокировками для мониторинга и оптимизации производительности базы данных.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Новиков Б. А. Основы технологий баз данных: учеб. пособие / Б. А. Новиков, Е. А. Горшкова, Н. Г. Графеева; под ред. Е. В. Рогова. — 2-е изд. — М.: ДМК Пресс, 2020. — 582 с. – URL: <https://postgrespro.ru/education/books/dbtech> (дата обращения: 20.02.2024)
2. Новиков Б. А. Лекции Основы технологий баз данных. – URL: <https://postgrespro.ru/education/university/dbtech> (дата обращения: 17.12.2023)
3. Моргунов, Е. П. PostgreSQL. Основы языка SQL: учеб. пособие / Е. П. Моргунов; под ред. Е. В. Рогова, П. В. Лузанова. — СПб.: БХВ-Петербург, 2018. — 336 с.: ил. – URL: <https://postgrespro.ru/education/books/sqlprimer> (дата обращения: 20.02.2024)
4. Моргунов, Е. П. Технологии разработки программ на основе инструментария с открытым исходным кодом. Вводный курс: учеб. пособие / Е. П. Моргунов, О. Н. Моргунова, В. В. Тынченко; НИИ СУВПТ. – Красноярск, 2006. – 148 с. – URL: <http://www.morgunov.org/docs/free_soft_tech.pdf> (дата обращения: 20.02.2024)
5. Лузанов П.В. и др. Postgres. Первое знакомство. – 178 с. – URL: <https://postgrespro.ru/education/books/introbook> (дата обращения: 20.02.2024)
6. BEGIN, COMMIT, ROLLBACK (работа с транзакциями) – URL: <https://ydb.tech/docs/ru/postgresql/statements/begin_commit_rollback> (дата обращения: 20.02.2024)
7. Postgres Pro Standard – Явные блокировки – URL: <https://postgrespro.ru/docs/postgresql/16/explicit-locking> (дата обращения 12.03.2024)
8. Блокировки в PostgreSQL: 2. Блокировки строк – Erogov (Habr) – URL: <https://habr.com/ru/companies/postgrespro/articles/463819/> (дата обращения 12.03.2024)
9. Блокировки в PostgreSQL: 3. Блокировки других объектов – Erogov (Habr) – URL: <https://habr.com/ru/companies/postgrespro/articles/465263/> (дата обращения 12.03.2024)