

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 №8**

**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. Смоделировать ситуацию обновления одной и той же строки тремя командами UPDATE в разных сеансах. Изучить возникшие блокировки в представлении pg\_locks и убедиться, что все они понятны.
2. Воспроизвести взаимоблокировку трех транзакций. Можно ли разобраться в ситуации постфактум, изучая журнал сообщений?
3. Могут ли две транзакции, выполняющие единственную команду UPDATE одной и той же таблицы, заблокировать друг друга? Попробовать воспроизвести такую ситуацию.

**РЕФЕРАТ**

Белослудцев Е.Д., Практическая работа направления подготовки «Программная инженерия» на тему «Транзакции. Блокировки. Блокировки строк»: М. 2024 г., МИРЭА – Российский технологический университет (РТУ МИРЭА), Институт информационных технологий (ИИТ), кафедра инструментального и прикладного программного обеспечения (ИиППО) – 16 стр., 14 рис., 10 источн.

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

Целью работы является изучение основ транзакционной обработки данных, механизмов управления параллелизмом и использования блокировок строк для управления доступом к данным.

Belosludtsev E.D., Practical work of the direction of training “Software Engineering” on the theme “Transactions. Locks. Row locks”: M. 2024, MIREA – Russian Technological University (RTU MIREA), Institute of Information of Instrumental and Applied Software (IAPS) – 16 p., 14 fig., 10 sources.

Keywords: POSTGRESQL, DATABASE, TABLE, SQL, TRANSACTION, LOCK.

The aim of the work is to learn the basics of transactional data processing, concurrency control mechanisms and the use of string locks to control access to data.

**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](#_Toc163225075)

[ХОД РАБОТЫ 8](#_Toc163225076)

[ЗАКЛЮЧЕНИЕ 14](#_Toc163225077)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 15](#_Toc163225078)

# ВВЕДЕНИЕ

В мире реляционных баз данных, транзакции играют фундаментальную роль в обеспечении надежности и целостности данных. Транзакция представляет собой единицу логической работы, которая включает в себя одну или несколько операций базы данных, гарантирующих их полное выполнение или полную отмену в случае сбоя или ошибки. Это обеспечивает согласованность данных в базе в любой момент времени, поддерживая их надежность и целостность.

Однако для того, чтобы гарантировать правильное выполнение транзакций и предотвращать возможные конфликты при параллельном доступе к данным из различных транзакций, применяются механизмы блокировок. Блокировки используются для контроля доступа к данным, обеспечивая, что только одна транзакция может изменять определенные данные в определенный момент времени.

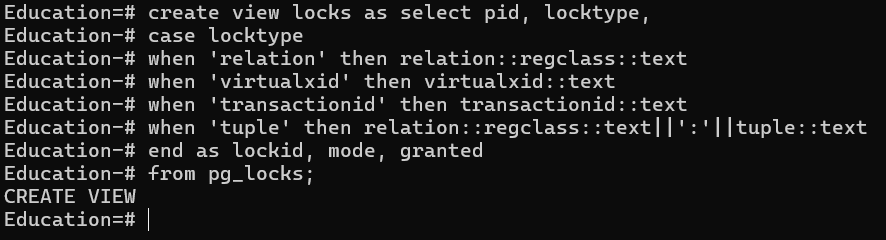
Существует два основных типа блокировок:

1. Блокировки объектов: Эти блокировки применяются к целым объектам базы данных, таким как таблицы или страницы данных. Они гарантируют, что только одна транзакция может иметь доступ к объекту для чтения или записи в определенный момент времени, что предотвращает конфликты доступа.
2. Блокировки строк: Эти блокировки применяются на уровне отдельных строк данных внутри таблицы. Они позволяют более гибко управлять параллелизмом, разрешая одновременный доступ к разным строкам одной таблицы из разных транзакций.

Эффективное управление транзакциями и блокировками имеет критическое значение для поддержания надежности и целостности данных, а также для обеспечения согласованного состояния базы данных в условиях параллельного доступа. Безопасность транзакций и уровень изоляции, обеспечиваемый блокировками, являются ключевыми аспектами современных систем управления базами данных, гарантирующими их корректное функционирование в реальном мире.

# ХОД РАБОТЫ

Приступим к выполнению первого задания. Для этого в самом начале создадим представления «locks». Данное представление будет содержать информацию о блокировках, которые существуют в данный момент времени в базе данных – рисунок 1.

  
Рисунок 1 – Создание представления над pg\_locks

Следующим шагом будет запуск работы первой транзакции, предварительно выполним выборку данных на примере таблицы «product», а также внесения изменений в одну из строк данной таблицы – рисунок 2.

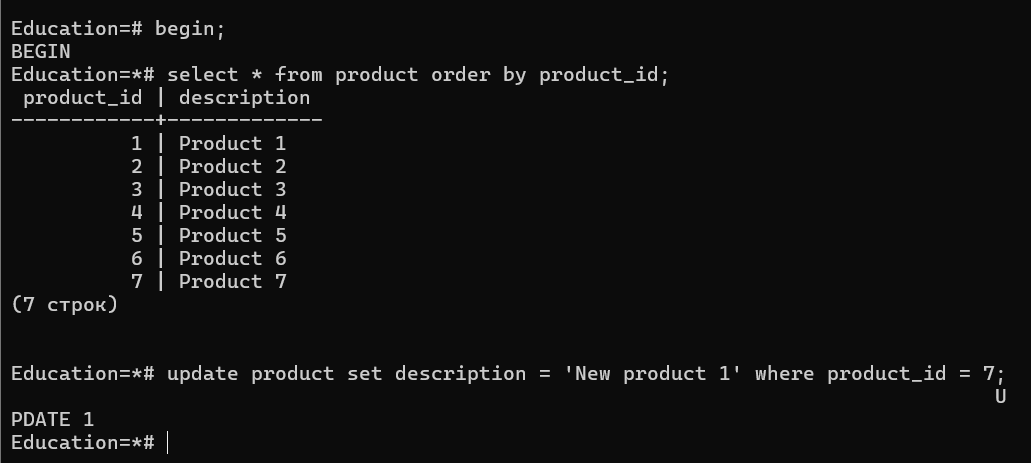


Рисунок 2 – Первоначальная выборка и обновление строки (первая транзакция)

Повторим предыдущий шаг для второй транзакции, предварительно выполнив выборку идентификатора транзакции и идентификатора процесса- рисунок 3.

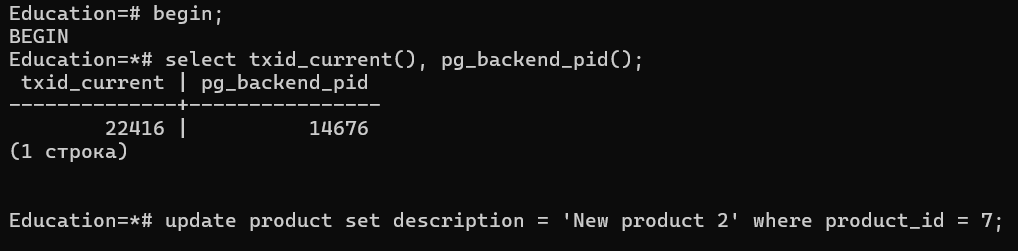


Рисунок 3 – Получение ID и обновление строки (вторая транзакция)

Аналогичные шаги выполняем с третьей транзакцией – рисунок 3.

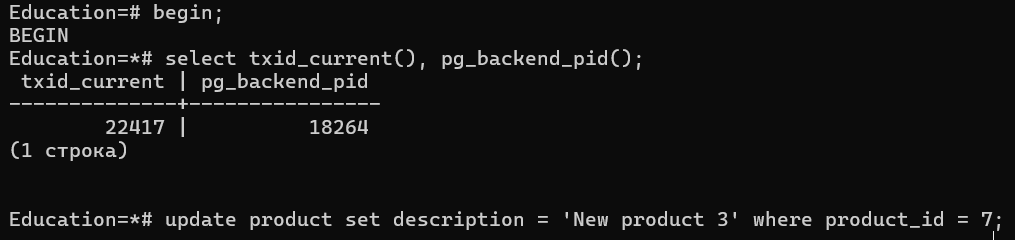


Рисунок 4 – Получение ID и обновление строки (третья транзакция)

Следующим шагом будет просмотр содержимое представления «locks» для каждого процесса – рисунки 5 и 6.

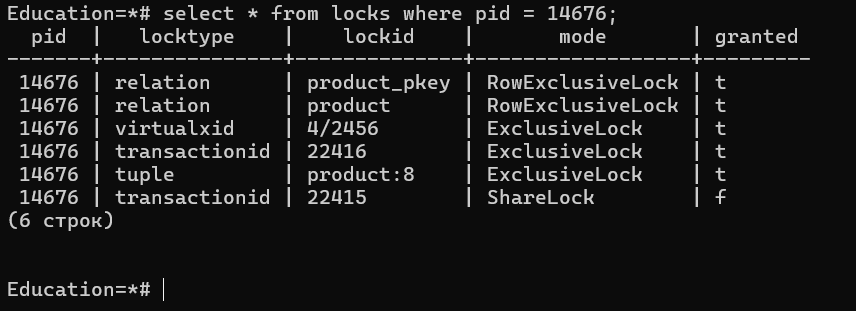


Рисунок 5 – Представление pg\_locks (pid = 14676)

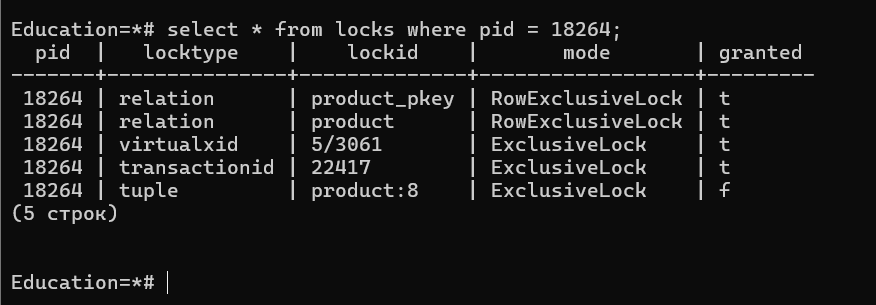


Рисунок 6 – Представление pg\_lock (pid = 18264)

Первая транзакция уже внесла изменения в строку таблицы. Вторая транзакция пытается обновить ту же строку, поэтому ей нужно получить разделяемую блокировку, чтобы убедиться, что она "видит" изменения, внесенные первой транзакцией. Однако вторая транзакция ожидает завершения первой, поскольку она еще не завершилась. Третья транзакция также пытается обновить эту строку, но она также не может этого сделать из-за блокировок, наложенных другими транзакциями. Для завершения этих транзакций необходимо сначала зафиксировать первую, затем вторую и, наконец, третью.

Вторым заданием данной практической работы было воспроизвести взаимоблокировку трех транзакций и пытаться разобраться в полученной ситуации, изучая журнал сообщений.

Первым шагом будет начало первой транзакции и изменения строки данных в таблице «product» – рисунок 7.

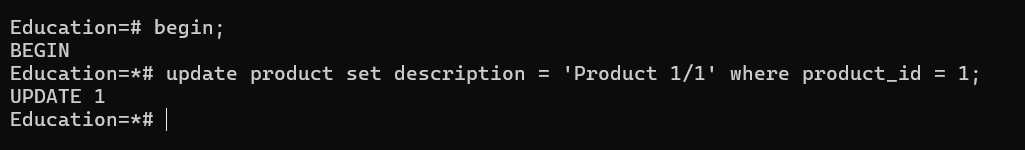


Рисунок 7 – Изменение первой строки (первая транзакция)

Теперь запускаем в работу вторую транзакцию, изменяем данные в таблице «product», которые были захвачены уже первой транзакции – рисунок 8.

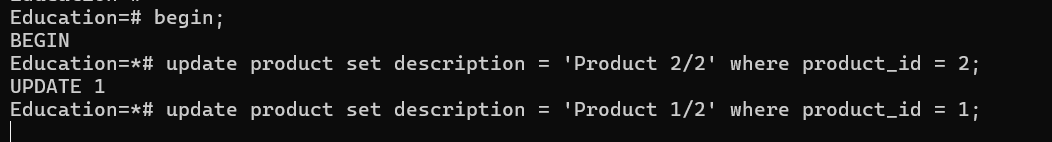


Рисунок 8 – Изменение второй строки и попытка изменения первой (вторая транзакция)

Выполняем аналогичные действия только для третьей транзакции (пытаемся изменить данные, которые были захвачены второй транзакцией) – рисунок 9.

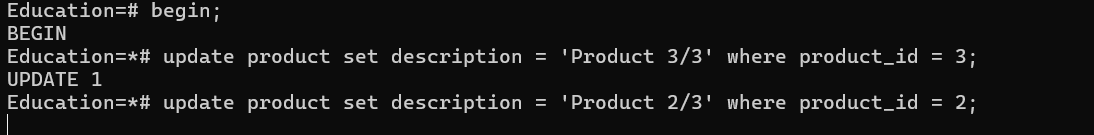


Рисунок 9 – Изменение третьей строки и попытка изменения второй (третья транзакция)

Возвращаемся к первой транзакции и пытаемся изменить данные, которые были захвачены третьей транзакцией – рисунок 10.

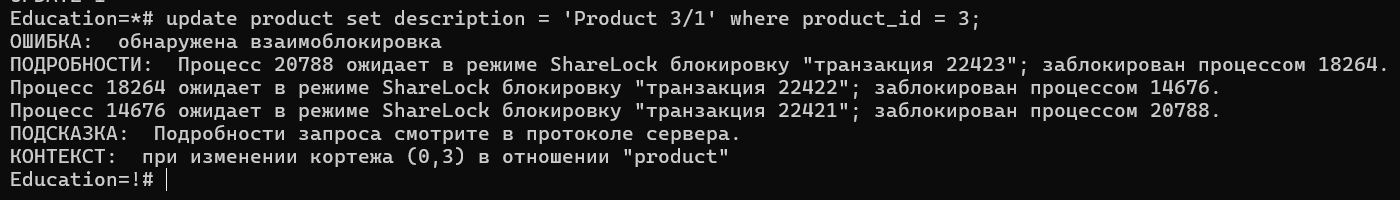


Рисунок 10 – Попытка изменения третьей строки (первая транзакция)

В данном сценарии происходит циклическая блокировка, где транзакция 3 блокирует транзакцию 2, которая в свою очередь блокирует транзакцию 1, а затем транзакция 1 блокирует транзакцию 3. Это приводит к зацикливанию блокировок, что создает потенциальную опасность для зависания системы. В таких случаях система обнаруживает этот цикл и прерывает одну из транзакций для разрыва цикла. В данном случае была прервана первая транзакция. Это изображено на рисунке 11 в логах журнала сообщений.

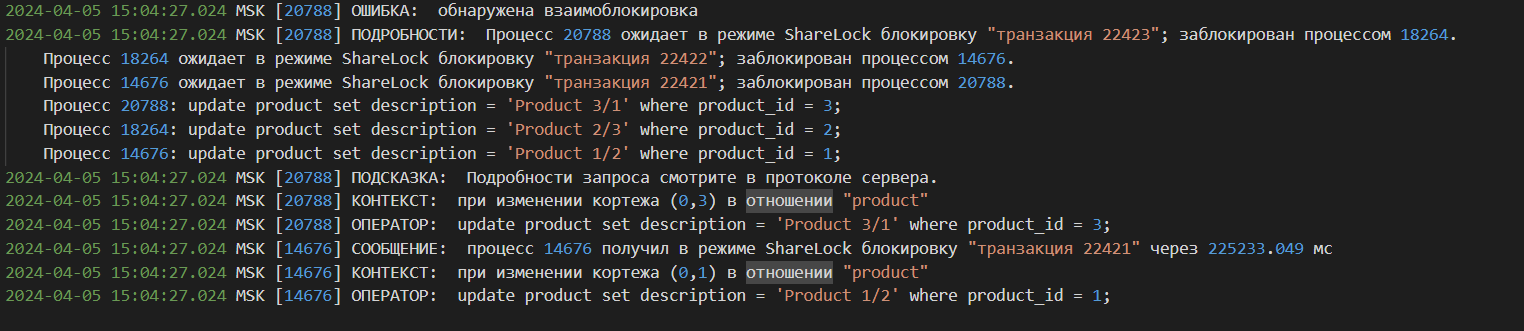


Рисунок 11 – Логи журнала сообщений

Да, из анализа журнала сообщений взаимоблокировки можно попытаться разобраться в ситуации, чтобы выявить причины и найти способы ее разрешения.

Это создает замкнутую цепь взаимоблокировки. Каждый процесс ожидает блокировку, удерживаемую следующим процессом в цепи.

Из этой информации можно выявить, какие транзакции и какие операции привели к взаимоблокировке. В данном случае, каждая транзакция пытается обновить строку в таблице "product", но они делают это в разном порядке, что приводит к взаимной блокировке.

При фиксации первой транзакции ожидаемо происходит откат, который позволяет второй транзакции завершить обработку первой строки, а третья транзакция также ждет снятия с блокировки на вторую строку, установленной второй транзакцией (рис. 12-14).



Рисунок 12 – Фиксация первой транзакции

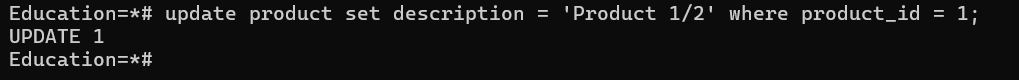


Рисунок 13 – Вывод во второй транзакции

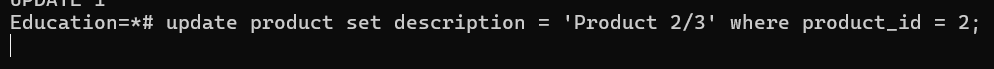


Рисунок 14 – Вывод в третьей транзакции

Ситуация, в которой две транзакции, выполняющие единственную (атомарную) команду UPDATE на одной и той же таблице, блокируют друг друга, невозможна. В данной ситуации первая транзакция сможет изменить и первоначально заблокировать строку (поскольку строка до этого не была заблокирована никакой другой транзакцией), тогда как вторая транзакция при попытке изменения будет ожидать разрешения блокировки. Это не приведет к возникновению циклического ожидания.

На рисунке 7 данной ситуации иллюстрируется следующим образом: первая транзакция успешно изменяет и блокирует строку, а вторая транзакция ожидает разрешения на блокировку, но не вызывает циклического ожидания.

# ЗАКЛЮЧЕНИЕ

В ходе данной работы были изучены основы транзакционной обработки данных, механизмы управления параллелизмом и использование блокировок строк для контроля доступа к данным. Мы смоделировали ситуации взаимной блокировки двух и трех транзакций, а также с помощью журнала сообщений получили информацию о идентификаторах процессов, участвовавших в блокировках, типах блокировок и идентификаторах объектов, на которые они были установлены.

В результате работы нам удалось глубже понять механизмы блокировок и взаимодействия транзакций, что помогает оптимизировать работу с данными и эффективно устранять блокировки в базе 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)
10. Почему в базе данных происходит взаимоблокировка? – Андрей Шагин (Medium) – URL: <https://medium.com/nuances-of-programming/почему-в-базе-данных-происходит-взаимоблокировка-c424ccb66ac5>