|  |  |  |
| --- | --- | --- |
|  |  |  |
| 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 №8**

**по дисциплине**

«Технологии обработки транзакций клиент-серверных приложений»

Выполнил студент группы ИКБО-20-21 Фомичев Р.А.

Принял Маличенко С.В.

Москва 2024**ЗАДАНИЕ НА ПРАКТИЧЕСКУЮ РАБОТУ**

1. Смоделировать ситуацию обновления одной и той же строки тремя командами UPDATE в разных сеансах. Изучить возникшие блокировки в представлении pg\_locks.
2. Воспроизвести взаимоблокировку трех транзакций.
3. Попробовать воспроизвести ситуацию выполнения UPDATE одной таблицы двумя транзакциями.

**РЕФЕРАТ**

Отчёт 17 страниц, 16 рисунков, 5 источников, 3 приложения.

POSTGRESQL, ТРАНЗАКЦИИ, БЛОКИРОВКИ, БЛОКИРОВКИ СТРОК, SQL

Объектом разработки является база данных.

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

В процессе работы производилось изучение блокировок строк.

Результатом являются сведения по механизмам действия блокировок строк.

**ВВЕДЕНИЕ**

[ВВЕДЕНИЕ 6](#_Toc19262)

[1. ВЫПОЛНЕНИЕ ПРАКТИЧЕСКОЙ РАБОТЫ 7](#_Toc16725)

[1.1 Выполнения задания 1 7](#_Toc27383)

[1.2 Выполнение задания 2 9](#_Toc20496)

[1.3 Выполнение задания 3 10](#_Toc10681)

[ЗАКЛЮЧЕНИЕ 12](#_Toc13588)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 13](#_Toc28781)

[ПРИЛОЖЕНИЕ А 14](#_Toc29393)

**ПЕРЕЧЕНЬ СОКРАЩЕНИЙ И ОБОЗНАЧЕНИЙ**

БД – база данных;

СУБД – система управления базами данных.

**ВВЕДЕНИЕ**

В современном информационном обществе, где огромные объёмы данных являются ключевым ресурсом для бизнеса, науки и повседневной жизни, вопрос обеспечения безопасности и надёжности хранения данных становится более критическим, чем когда-либо. Одним из основных аспектов обеспечения безопасности и надёжности данных в СУБД является блокировки отношений и других объектов.

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

1. **ВЫПОЛНЕНИЕ ПРАКТИЧЕСКОЙ РАБОТ****Ы**

**1.1 Выполнения задания 1**

Для выполнения практической работы была создана и заполнена данными таблица example\_table. Содержимое таблицы представлено на рисунке 1.

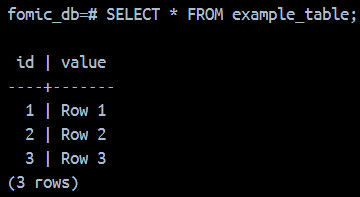


Рисунок 1 – Содержимое example\_table

Затем в первом сеансе была начата транзакция для изменения строки с id 1. Результат выполнения команды UPDATE представлен на рисунке 2.

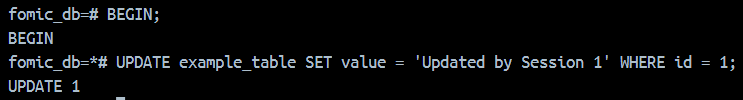


Рисунок 2 – Изменение строки в первом сеансе

Был начат второй сеанс, в котором также была начата новая транзакция для изменения строки с id 1. Результат выполнения команды UPDATE представлен на рисунке 3.

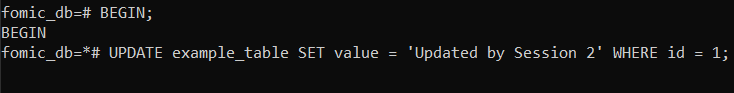


Рисунок 3 – Изменение строки во втором сеансе

Был начат третий сеанс, в котором также была начата новая транзакция для изменения строки с id 1. Результат выполнения команды UPDATE представлен на рисунке 4.

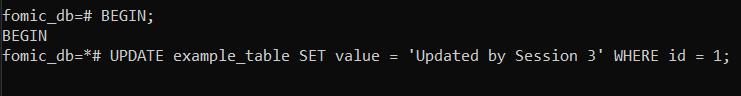


Рисунок 4 – Изменение строки в третьем сеансе

Транзакции 2 и 3 ожидают выполнения первой. Выполнение команды pg\_locks представлено на рисунке 5.

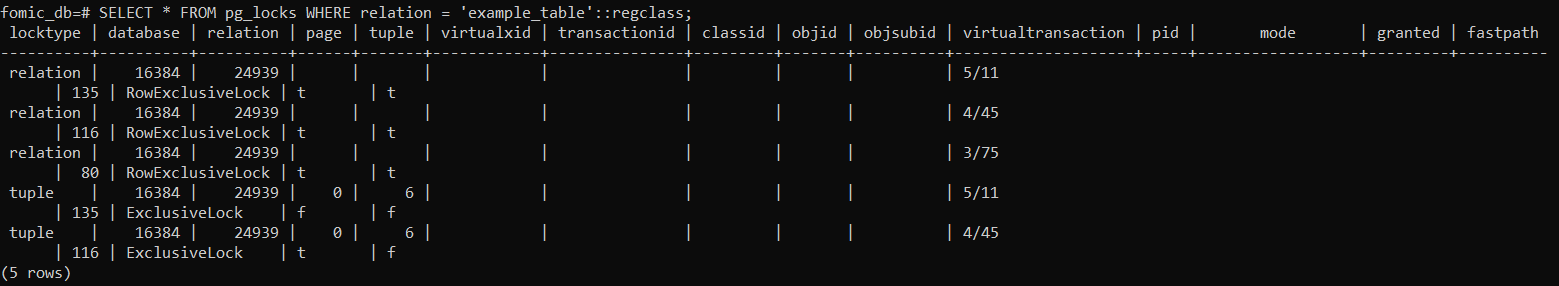


Рисунок 5 – Возникшие блокировки

## 1.2 Выполнение задания 2

Таблица была приведена к начальному состоянию. Содержимое таблицы example\_table представлено на рисунке 6.

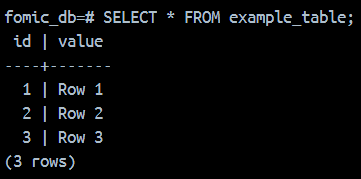


Рисунок 6 – Содержимое example\_table

В первом сеансе была начата транзакция по изменению первой строки таблицы. Результат использования команды UPDATE представлен на рисунке 7.

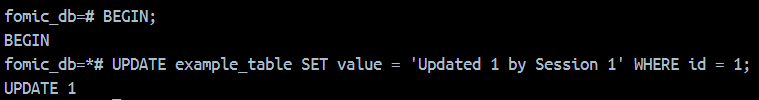


Рисунок 7 – Изменение первой строки в первом сеансе

Во втором сеансе была начата транзакция по изменению второй строки таблицы. Результат использования команды UPDATE представлен на рисунке 8.

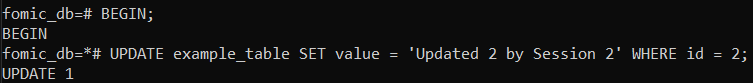


Рисунок 8 – Изменение второй строки во втором сеансе

В третьем сеансе была начата транзакция по изменению третьей строки таблицы. Результат использования команды UPDATE представлен на рисунке 9.

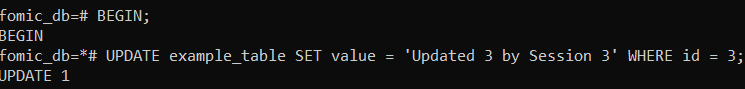


Рисунок 9 – Изменение второй строки во втором сеансе

В первом сеансе была произведена попытка по изменению второй строки. Результат использования команды UPDATE представлен на рисунке 10.

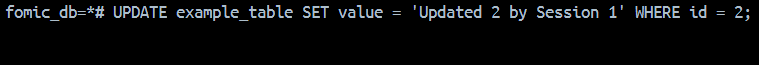


Рисунок 10 – Попытка изменения второй строки в первом сеансе

Во втором сеансе была произведена попытка по изменению третьей строки. Результат использования команды UPDATE представлен на рисунке 11.



Рисунок 11 – Попытка изменения первой строки в первом сеансе

В третьем сеансе была произведена попытка по изменению первой строки. Результат использования команды UPDATE представлен на рисунке 12.



Рисунок 12 – Попытка изменения первой строки в первом сеансе

Возникает циклическое ожидание, который никогда не завершится само по себе. Поэтому сервер, обнаружив такой цикл, прерывает одну из транзакций. Результат прерывания цикла представлен на рисунке 13.

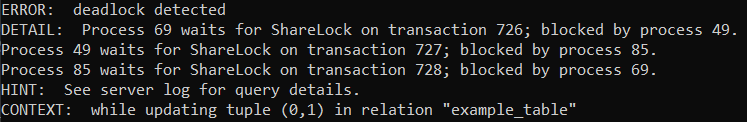


Рисунок 13 – Прерывание транзакции

Третья транзакция была прервана, из-за чего вторая была успешно выполнена. Содержимое таблицы example\_table представлено на рисунке 14.

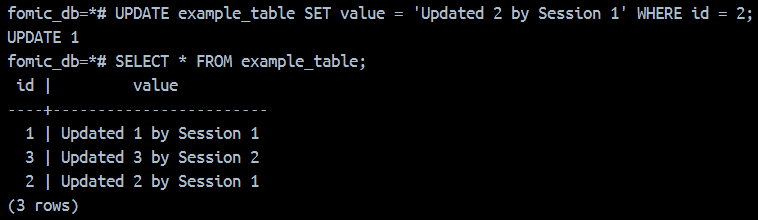


Рисунок 14 – Содержимое example\_table

## 1.3 Выполнение задания 3

Была создана и заполнена новая таблица. Результат представлен на рисунке 15.

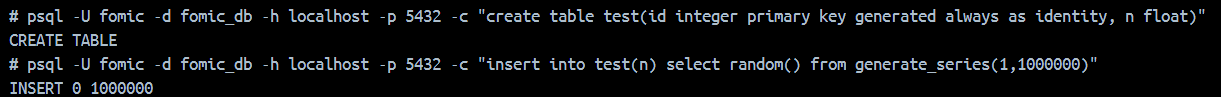


Рисунок 15 – Создание и заполнение новой таблицы

Затем, почти одновременно, были начаты транзакции в первой и во второй сессиях. В первой сессии строки изменяются по возрастанию id, а во второй по убыванию. Результат выполнения транзакций представлен на рисунках 16, 17.

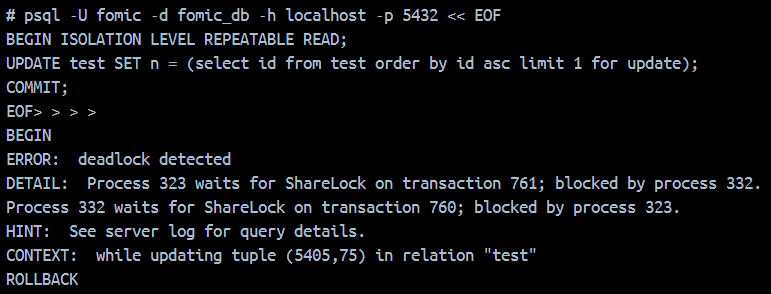


Рисунок 16 – Изменение строк таблицы в первой сессии

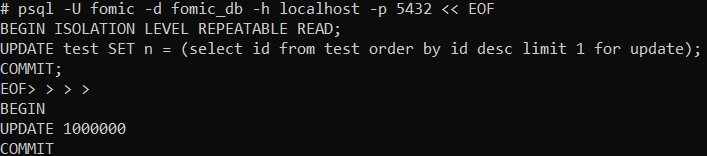


Рисунок 17 – Изменение строк таблицы во второй сессии

Спустя определенное время, транзакции «пересеклись» и создали взаимоблокировку. Первая транзакция была прервана. Результат прерывания представлен на рисунке 18.

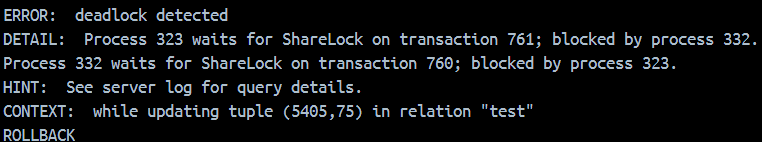


Рисунок 18 – Прерывание первой транзакции

Таким образом, можно сделать вывод, что две транзакции, выполняющие одну команду UPDATE одной и той же таблицы могут заблокировать друг друга.

# ЗАКЛЮЧЕНИЕ

В ходе выполнения практической работы были изучены основы транзакционной обработки данных, механизмы управления параллелизмом и использования блокировок строк для управления доступом к данным.

# 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)

# ПРИЛОЖЕНИЕ А

**ФРАГМЕНТЫ КОДА РАЗРАБОТАННОГО ПРИЛОЖЕНИЯ**

Листинг 1 – SQL код для выполнения задания 1 практической работы

-- Session 1 --

CREATE TABLE example\_table (

id SERIAL PRIMARY KEY,

value VARCHAR(50)

);

INSERT INTO example\_table (value) VALUES

('Row 1'),

('Row 2'),

('Row 3');

BEGIN;

UPDATE example\_table SET value = 'Updated by Session 1' WHERE id = 1;

-- Session 2 --

BEGIN;

UPDATE example\_table SET value = 'Updated by Session 2' WHERE id = 1;

-- Session 3 --

BEGIN;

UPDATE example\_table SET value = 'Updated by Session 3' WHERE id = 1;

-- Session 4 --

SELECT \* FROM pg\_locks WHERE relation = 'example\_table'::regclass;

Листинг 2 – SQL код для выполнения задания 2 практической работы

-- Session 1 --

BEGIN;

UPDATE example\_table SET value = 'Updated 1 by Session 1' WHERE id = 1;

-- Session 2 --

BEGIN;

UPDATE example\_table SET value = 'Updated 2 by Session 2' WHERE id = 2;

-- Session 3 --

BEGIN;

UPDATE example\_table SET value = 'Updated 3 by Session 3' WHERE id = 3;

-- Session 1 --

UPDATE example\_table SET value = 'Updated 2 by Session 1' WHERE id = 2;

-- Session 2 --

UPDATE example\_table SET value = 'Updated 3 by Session 2' WHERE id = 3;

-- Session 3 --

UPDATE example\_table SET value = 'Updated 1 by Session 3' WHERE id = 1;

Листинг 3 - SQL код для выполнения задания 3 практической работы

-- Session 1 --

BEGIN;

UPDATE example\_table SET value = 'Updated by Session 1' WHERE id = 1;

-- Session 2 --

BEGIN;

UPDATE example\_table SET value = 'Updated by Session 2' WHERE id = 1;