|  |  |  |
| --- | --- | --- |
|  |  |  |
| 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 №7**

**по дисциплине**

«Технологии обработки транзакций клиент-серверных приложений»

Выполнил студент группы ИКБО-20-21 Сидоров С. Д.

Принял Маличенко С.В.

Москва 2024

**ЗАДАНИЕ НА ПРАКТИЧЕСКУЮ РАБОТУ**

1. Какие блокировки на уровне изоляции Read Committed удерживает транзакция, прочитавшая одну строку таблицы по первичному ключу? Проверьте на практике.
2. Повторите предыдущий пункт для уровня изоляции Serializable.
3. Настройте сервер так, чтобы в журнал сообщений сбрасывалась информация о блокировках, удерживаемых более 100 миллисекунд. Воспроизведите ситуацию, при которой в журнале появятся такие сообщения.

**РЕФЕРАТ**

Отчёт 16 страниц, 12 рисунков, 5 источников, 1 приложение.

POSTGRESQL, ТРАНЗАКЦИИ, БЛОКИРОВКИ, БЛОКИРОВКИ ОБЪЕКТОВ, SQL

Объектом разработки является база данных.

Цель работы – исследование типов блокировок.

В процессе работы производилось изучение различных типов блокировок.

Результатом являются сведения по механизмам действия различных типов блокировок.

**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 исследование различных типов блокировок и механизмов их действия с целью оптимизации использования ресурсов.

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

**1.1 Выполнения задания 1**

Для выполнения первого задания практической работы была создана база данных locks\_objects внутри которой, создана таблица accounts содержащая три строки, что продемонстрировано на рисунке 1.

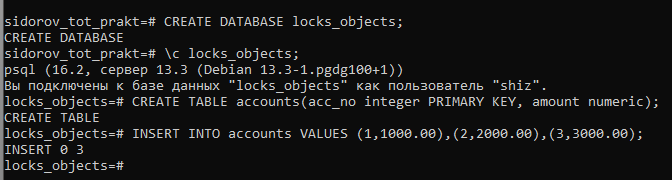


Рисунок 1 – Подготовка базы данных

Для выполнения первого задания была запущена вторая консоль, в которой был получен текущий pg\_backend\_pid и начата новая транзакция с уровнем изоляции Read Committed, которая читает одну строку из таблицы accounts, что представлено на рисунке 2.

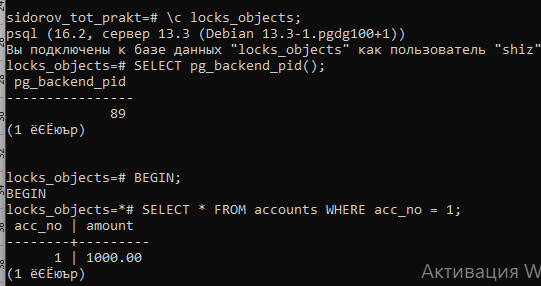


Рисунок 2 – Создание транзакции с уровнем изоляции Read Committed

Для получения информации об удерживаемых блокировках был создан запрос, результат которого продемонстрирован на рисунке 3.

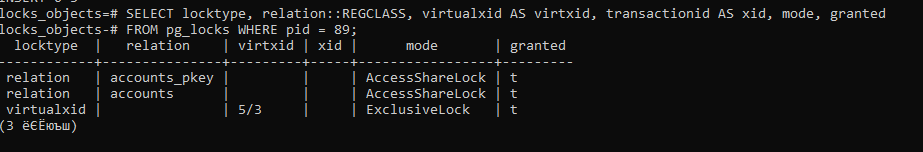


Рисунок 3 – Блокировки удерживаемые транзакцией с уровнем изоляции Read Committed

Результат получения списка блокировок внутри транзакции продемонстрирован на рисунке 4.

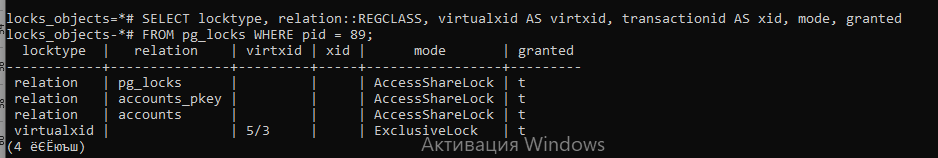


Рисунок 4 – Блокировки удерживаемые транзакцией с уровнем изоляции Read Committed полученные из самой транзакции

В результате список удерживаемых блокировок содержит:

* Блокировка таблицы accounts в режиме AccessShareLock;
* Блокировка индекса accounts\_pkey, созданного для первичного ключа, в том же режиме;
* Исключительная блокировка собственного номера виртуальной транзакции.

Внутри самой транзакции добавляется блокировка на таблицу pg\_locks.

## 1.2 Выполнение задания 2

После завершения предыдущей транзакции была начата новая транзакция с уровнем изоляции Serializable, в которой также был отправлен запрос на получение одной строки из таблицы accounts, что продемонстрировано на рисунке 5.

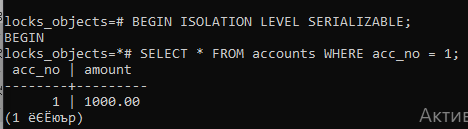


Рисунок 5 – Создание транзакции с уровнем изоляции Serializable

После начала транзакции в отдельной консоли был получен список блокировок, удерживаемых данной транзакцией, который представлен на рисунке 6.

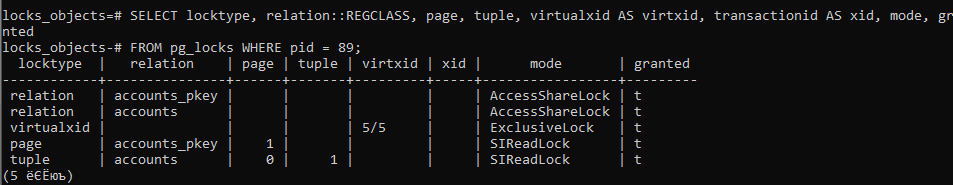


Рисунок 6 – Список блокировок, удерживаемых транзакцией с уровнем изоляции Serializable

Новый список блокировок содержит те же блокировки, что и предыдущая транзакция с уровнем изоляции Read Committed, а также добавляет предикатную блокировку страницы индекса и предикатную блокировку прочитанной версии строки.

## 1.3 Выполнение задания 3

Для выполнения данного задания была произведена СУБД, которая продемонстрирована на рисунке 7, в результате чего в журнал сообщений будет записываться информация о блокировках длящихся более 100 мс.

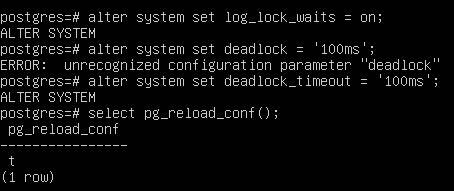
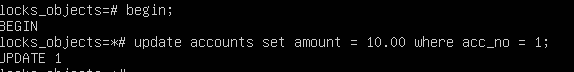


Рисунок 7 - Настройка СУБД согласно заданию

Для демонстрации примененных изменений были созданы две транзакции, каждая из которых обновляет одну и ту же строку в таблице accounts, после чего в первом сеансе используется применяется задержка с помощью pg\_sleep() и поочерёдно завершаются транзакции в обоих сеансах. Описанные выше действия продемонстрированы на рисунках 8 - 11.

  
Рисунок 8 - Начало транзакции в сеансе 1

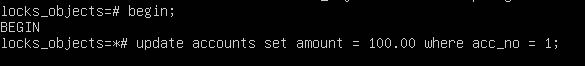


Рисунок 9 - Начало транзакции в сеансе 2

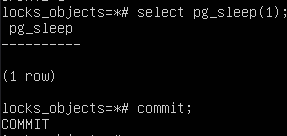


Рисунок 10 - Установка задержки и завершение транзакции в сеансе 1

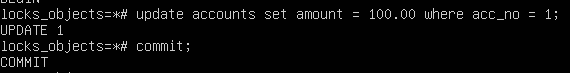


Рисунок 11 - Завершение транзакции в сеансе 2

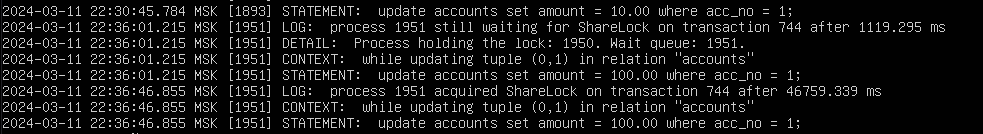
После завершения обоих транзакций, было получено содержимое журнала сообщений, которое продемонстрировано на рисунке 12.  


Рисунок 12 - Содержимое файла журнала сообщений

В журнале сообщений находится сообщения о блокировке Sharelock для транзакции 744, которое было помещено в файл за счёт выше перечисленных действий.

# ЗАКЛЮЧЕНИЕ

В ходе выполнения практической работы были получены знания о различных типах блокировок и механизмов их действия с целью оптимизации использования ресурсов. Также на практике были рассмотрены ситуации появления различных типов блокировок и механизмов отслеживания появления блокировок удерживаемых длительное время.

# 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

##Console 1

CREATE DATABASE locks\_objects;

\c locks\_objects;

CREATE TABLE accounts(acc\_no integer PRIMARY KEY, amount numeric);

INSERT INTO accounts VALUES (1,1000.00),(2,2000.00),(3,3000.00);

##Console 2

\c locks\_objects;

SELECT pg\_backend\_pid();

BEGIN;

SELECT \* FROM accounts WHERE acc\_no = 1;

##Console 1

SELECT locktype, relation::REGCLASS, virtualxid AS virtxid, transactionid AS xid, mode, granted

FROM pg\_locks WHERE pid = 89;

##Console 2

SELECT locktype, relation::REGCLASS, virtualxid AS virtxid, transactionid AS xid, mode, granted

FROM pg\_locks WHERE pid = 89;

COMMIT;

Листинг 2 – SQL код для выполнения задания 2 практической работы

##Console 2

SELECT pg\_backend\_pid();

BEGIN ISOLATION LEVEL SERIALIZABLE;

SELECT \* FROM accounts WHERE acc\_no = 1;

##Console 1

SELECT locktype, relation::REGCLASS, page, tuple, virtualxid AS virtxid, transactionid AS xid, mode, granted

FROM pg\_locks WHERE pid = 89;

##Console 2

Commit;

Листинг 3 - SQL код для выполнения задания 3 практической работы

##Проводим настройку

ALTER SYSTEM SET log\_lock\_waits = on;

ALTER SYSTEM SET deadlock\_timeout = '100ms';

SELECT pg\_reload\_conf();

##Воспроизводим блокировку

###Console 1

BEGIN;

UPDATE accounts SET amount = 10.00 WHERE acc\_no = 1;

###Console 2

BEGIN;

UPDATE accounts SET amount = 100.00 WHERE acc\_no = 1;

###Console 1

SELECT pg\_sleep(1);

COMMIT;

###Console 2

Commit;