

# Онлайн образование

otus.ru



Проверить, идет ли запись

**Меня хорошо видно  
&& слышно?**



Тема вебинара

# Блокировки



## Коробков Виктор

Консультант команды технологического обеспечения  
ООО «ИТ ИКС5 Технологии»

Telegram: @Korobkov\_Viktor



# Правила вебинара



Активно  
участвуем



Off-topic обсуждаем  
в Telegram



Задаем вопрос  
в чат или голосом



Вопросы вижу в чате,  
могу ответить не сразу

## Условные обозначения



Индивидуально



Время, необходимое  
на активность



Пишем в чат



Говорим голосом



Документ



Ответьте себе или  
задайте вопрос

# Маршрут вебинара



Общие понятия

Блокировки отношений

Блокировки строк

Блокировки других объектов

# Цели вебинара

После занятия вы сможете

1. Понимать как работают блокировки
  2. Находить проблемные места связанные с блокировками
- 
- 
-

# Общие понятия

# Блокировки объектов

Проблема - как одновременно получить доступ к одному объекту?



# Блокировки объектов

**Решение - упорядочение конкурентного доступа к разделяемым ресурсам**

- перед обращением к данным процесс захватывает блокировку
- после обращения — освобождает, другой процесс начинает ожидать освобождения ресурса
- блокировки приводят к очередям

**Дальнейшее развитие**

- многоверсионность — несколько версий данных, не избавляет от всех проблем
- оптимистичные блокировки — процессы не блокируются, но могут возникать ошибки

# Блокировки объектов

## Что подразумевается под ресурсом?

- все, что можно идентифицировать

## Примеры ресурсов

- реальные хранимые объекты: страницы, таблицы, строки и т. п.
- структуры данных в общей памяти (хеш-таблицы, буферы...)
- абстрактные ресурсы (число)

# Блокировки объектов

**Факторы, влияющие на производительность:**

**Гранулярность блокировки**

- степень детализации, уровень в иерархии ресурсов, например:  
таблица → страница → строки, хеш-таблица → корзины
- выше гранулярность — больше возможностей для параллелизма

**Режимы блокировок**

- совместимость режимов определяется матрицей
- больше совместимых режимов — больше возможностей для параллелизма

# Блокировки объектов

По типу:

- исключительные (exclusive)
- разделяемыми / совместные (shared)

# Блокировки объектов

**По времени жизни:**

## **Долговременные блокировки**

- обычно захватываются до конца транзакции и относятся к хранимым данным (например таблицы и отношения)
- большое число режимов
- развитая «тяжеловесная» инфраструктура, мониторинг

## **Краткосрочные блокировки**

- обычно захватываются на доли секунды (управляются автоматически)
- относятся к структурам в оперативной памяти
- минимум режимов
- «легковесная» инфраструктура, мониторинг может отсутствовать

# Блокировки объектов

## Информация в общей памяти сервера

представление pg\_locks:

- locktype — тип блокируемого ресурса,
- mode — режим блокировки

## Инфраструктура

- очередь ожидания: ждущие процессы не потребляют ресурсы
- обнаружение взаимоблокировок

## Ограниченное количество

`max_locks_per_transaction` × `max_connections`

при этом 1 транзакция может превысить `max_locks_per_transaction`

# Блокировки объектов

**Типы ресурсов** (столбцом locktype в pg\_locks):

- Relation — Блокировки отношений
- Transactionid — транзакция
- Virtualxid — виртуальная транзакция
- Tuple — версия строки
- Extend — добавление страниц к файлу отношения
- Object — не отношение: база данных, схема и т. п.
- Page — страница (используется некоторыми типами индексов)
- Advisory — рекомендательная блокировка

# Блокировки объектов

## Текущие блокировки

- представление `select * from pg_locks;`
- функция `select pg_blocking_pids(pid);`

## Вывод сообщений в журнал сервера

- параметр `SHOW log_lock_waits;`
- выводить сообщение об ожидании дольше `deadlock_timeout` или `log_min_duration_statement`

## Максимальное время ожидания блокировки

- `lock_timeout`



# Блокировки объектов: предикатные блокировки

**Задача:** реализация уровня изоляции Serializable

- используются в дополнение к обычной изоляции на снимках данных
- оптимистичные блокировки, название сложилось исторически

**Информация в общей памяти сервера**

- представление pg\_locks со специальным режимом SIReadLock

**Ограниченное количество**

- $\text{max\_pred\_locks\_per\_transaction} \times \text{max\_connections}$

**Повышение количества блокировок (автоэкскалация)**

- max\_pred\_locks\_per\_relation
- max\_pred\_locks\_per\_page

# Блокировки отношений

# Блокировки отношений

## Режимы

- Access Share
- Row Share
- Row Exclusive
- Share Update Exclusive

SELECT  
SELECT FOR UPDATE/SHARE  
UPDATE, DELETE, INSERT  
VACUUM, ALTER TABLE,  
CREATE INDEX CONCURRENTLY

- Share
- Share Row Exclusive
- Exclusive
- Access Exclusive

CREATE INDEX  
CREATE TRIGGER, ALTER TABLE  
REFRESH MAT. VIEW CONCURRENTLY  
DROP, TRUNCATE, VACUUM FULL,  
LOCK TABLE, ALTER TABLE, REFRESH MAT. VIEW

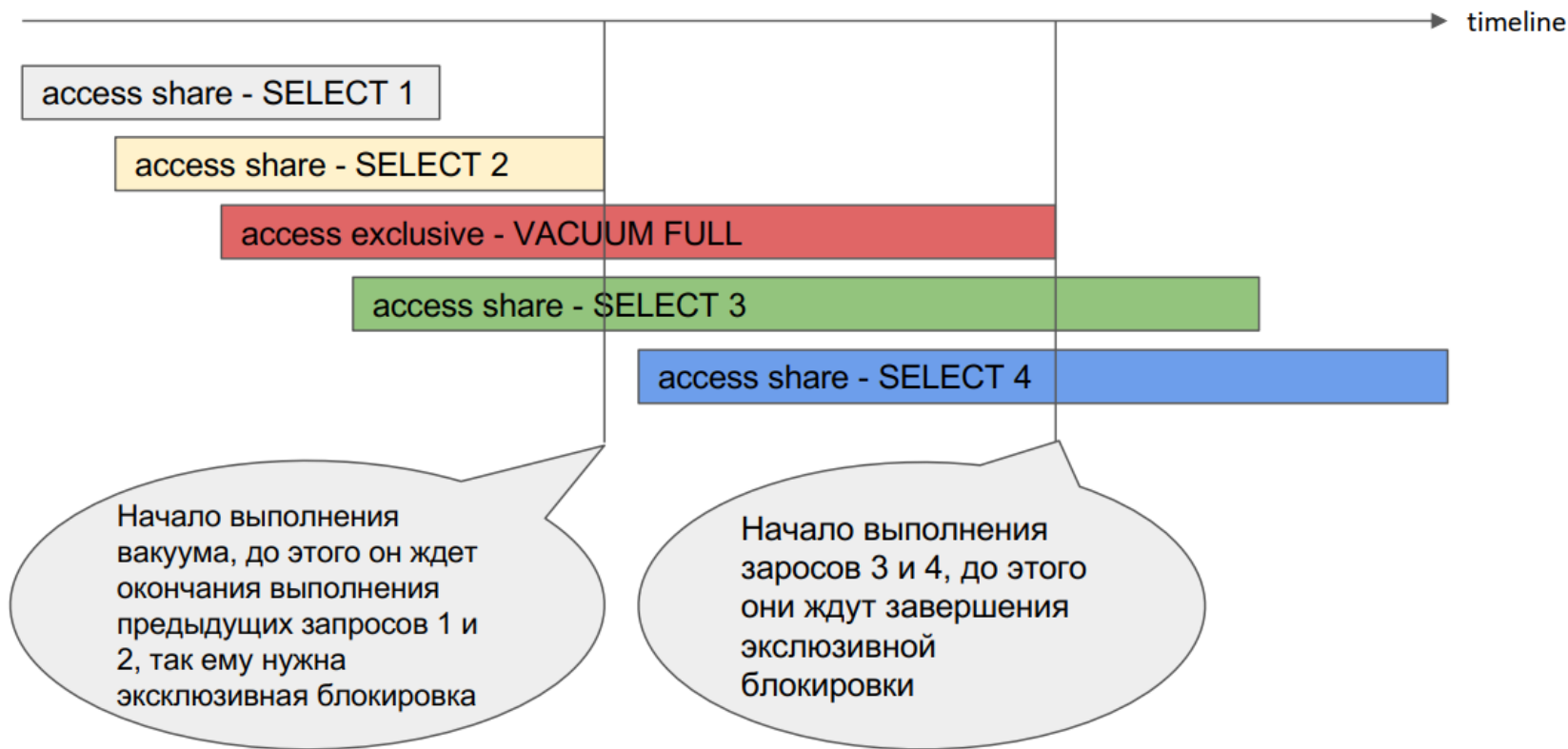


допускают  
изменение данных

# Блокировки отношений

Запрашиваемый режим блокировки	Текущий режим блокировки							
	ACCESS SHARE	ROW SHARE	ROW EXCLUSIVE	SHARE UPDATE EXCLUSIVE	SHARE	SHARE ROW EXCLUSIVE	EXCLUSIVE	ACCESS EXCLUSIVE
ACCESS SHARE								X
ROW SHARE							X	X
ROW EXCLUSIVE					X	X	X	X
SHARE UPDATE EXCLUSIVE				X	X	X	X	X
SHARE			X	X		X	X	X
SHARE ROW EXCLUSIVE			X	X	X	X	X	X
EXCLUSIVE		X	X	X	X	X	X	X
ACCESS EXCLUSIVE	X	X	X	X	X	X	X	X

# Блокировки отношений



# Блокировки строк

# Блокировки строк

- информация только в страницах данных
- поле xтах заголовка версии строки + информационные биты
- неограниченное количество (могут быть миллиарды строк)
- большое число минимально отражается на производительности
- очередь ожидания организована с помощью блокировок объектов (пропорционально числу процессов, а не строк)

# Блокировки строк

**Режимы:** Исключительный и разделяемый

**Исключительный:**

**UPDATE** - удаление строки или изменение всех полей

- SELECT FOR UPDATE
- UPDATE (с изменением ключевых полей)
- DELETE

**NO KEY UPDATE** - изменение любых полей, кроме ключевых

- SELECT FOR NO KEY UPDATE
- UPDATE (без изменения ключевых полей - все внешние ключи без изменений)



# Блокировки строк

**Режимы:** Исключительный и разделяемый

**Разделяемый:**

**SHARE** - запрет изменения любых полей строки

- SELECT FOR SHARE

**KEY SHARE** - запрет изменения ключевых полей строки

- SELECT FOR KEY SHARE
- UPDATE (без изменения ключевых полей)

# Блокировки строк

Запрашиваемый режим блокировки	Текущий режим блокировки			
	FOR KEY SHARE	FOR SHARE	FOR NO KEY UPDATE	FOR UPDATE
FOR KEY SHARE				X
FOR SHARE			X	X
FOR NO KEY UPDATE		X	X	X
FOR UPDATE	X	X	X	X

- исключительные режимы конфликтуют между собой;
- разделяемые режимы совместимы между собой;
- разделяемый режим FOR KEY SHARE совместим с исключительным режимом FOR NO KEY UPDATE (то есть можно одновременно обновлять неключевые поля и быть уверенным в том, что ключ не изменится).

# Блокировки строк

## Структура строки в **Исключительном** режиме

1000	xmin
1001	xmax (номер блокирующей транзакции)
t	xmin_committed
	xmin_aborted
	xmax_committed
	xmax_aborted
	xmax_lock_only (select for update... - данные еще актуальны, хоть и заблокированы)
	keys_updated (изменились ли значения ключевых полей)
(0,1)	ctid (физическое местоположение строки в таблице)
мама мыла раму	данные

# Блокировки строк

Структура строки в **Разделяющем режиме**, когда пришла вторая транзакция

1000	xmin
???	xmax (номер блокирующей транзакции)
t	xmin_committed
	xmin_aborted
	xmax_committed
	xmax_aborted
	xmax_lock_only (select for update...)
	keys_updated (изменились ли значения ключевых полей)
(0,1)	ctid (физическое местоположение строки в таблице)
мама мыла раму	данные



# Блокировки строк

## Структура строки в Разделяющем режиме

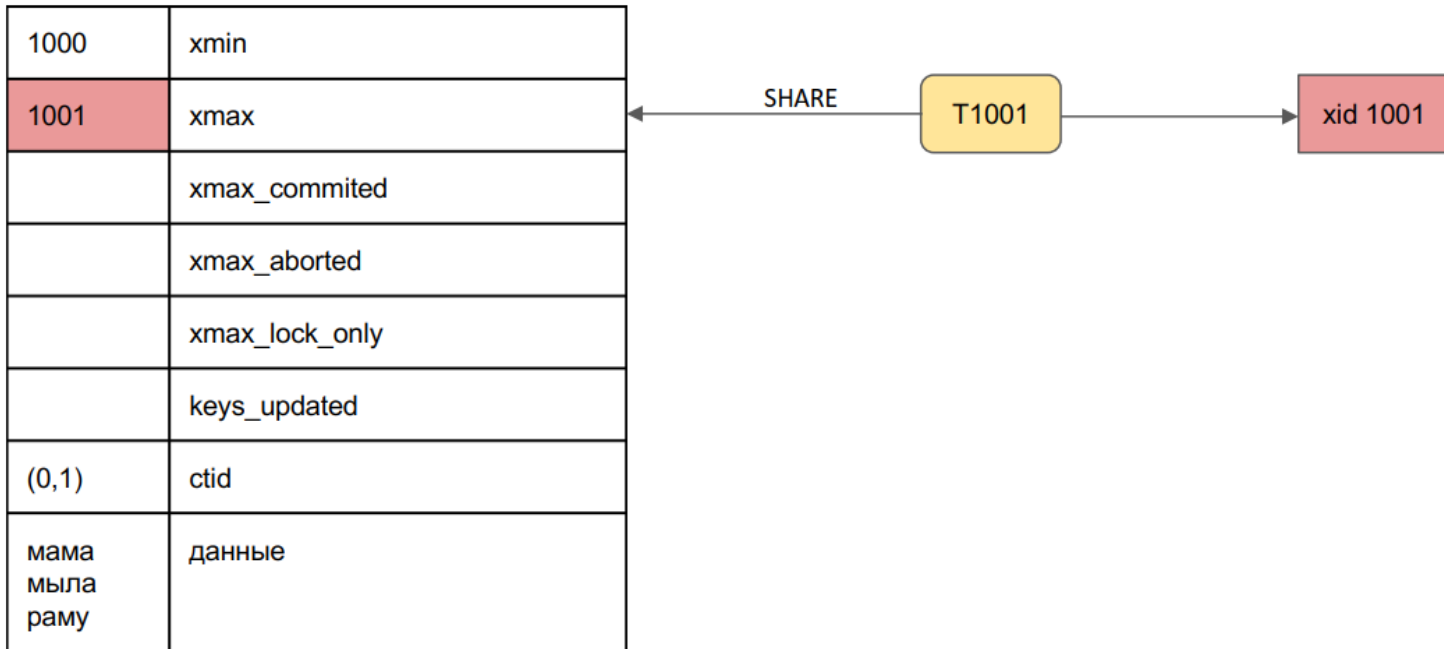
1000	xmin
2	xmax (номер блокирующей <b>МУЛЬТИ</b> транзакции)
t	xmin_committed
	xmin_aborted
	xmax_committed
t	xmax_is_multi
(0,1)	ctid
мама мыла раму	данные

\$PGDATA/pg\_multixact

1001	KEY SHARE
1020	NO KEY UPDATE

# Блокировки строк

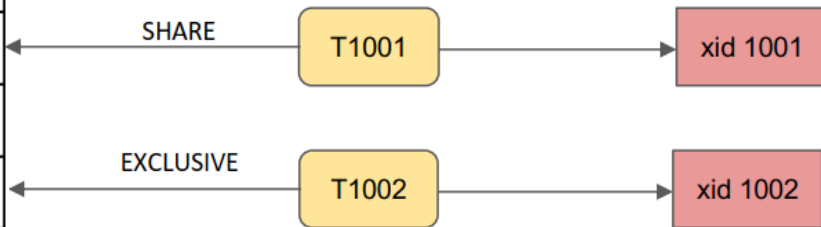
Как это происходит



# Блокировки строк

Как это происходит

1000	xmin
1001	xmax
	xmax_committed
	xmax_aborted
	xmax_lock_only
	keys_updated
(0,1)	ctid
мама мыла раму	данные

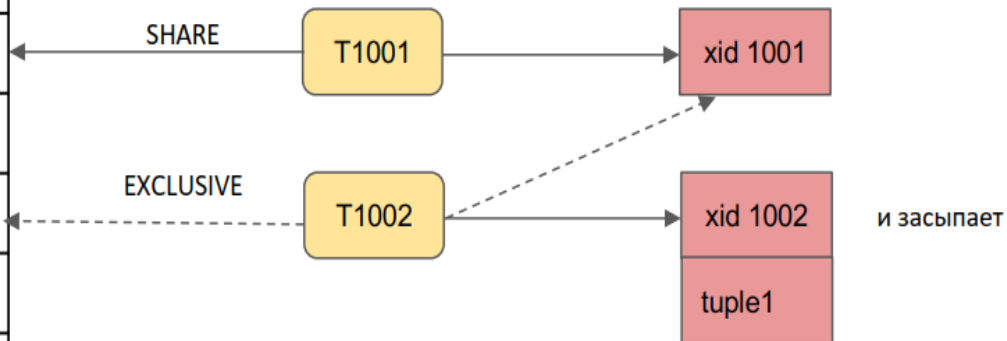


что произойдет дальше?

# Блокировки строк

Как это происходит

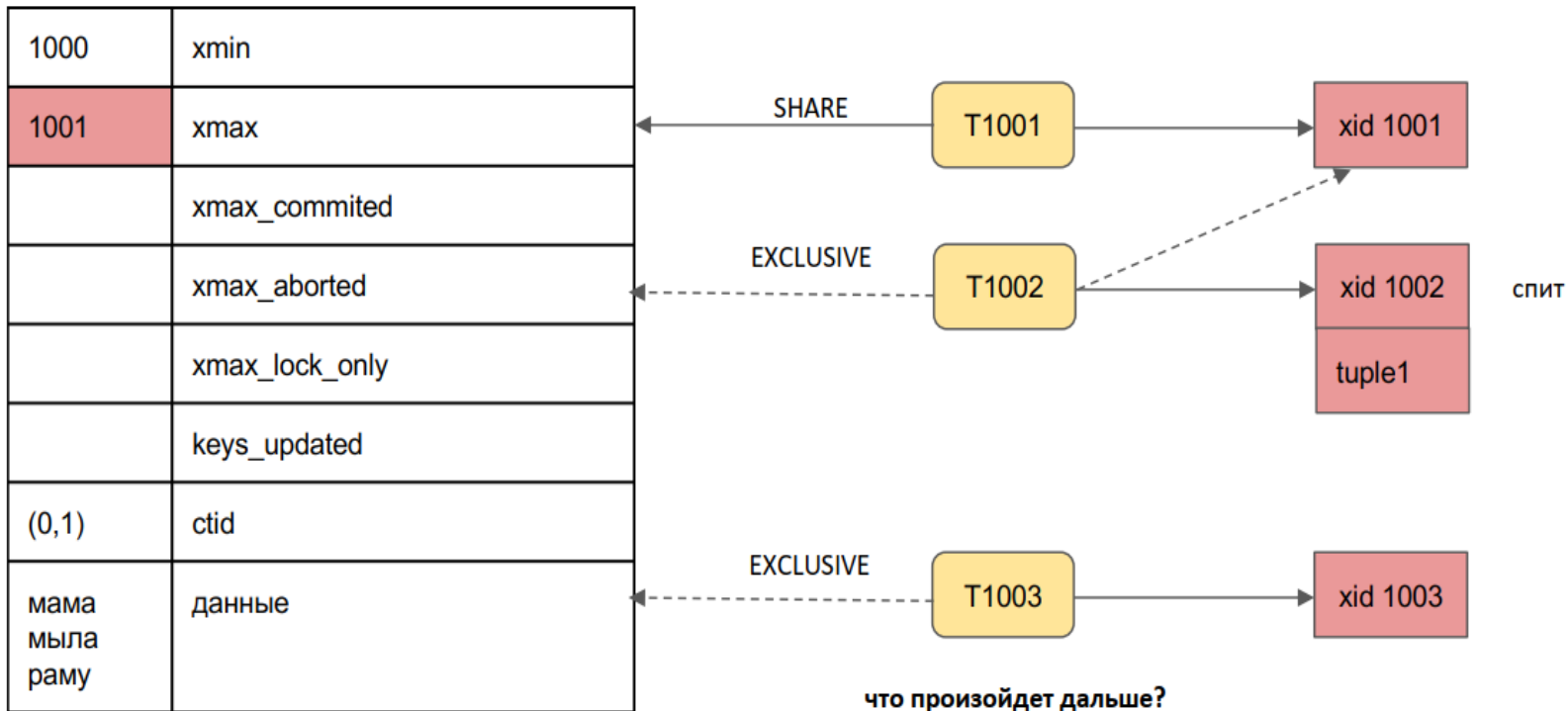
1000	xmin
1001	xmax
	xmax_committed
	xmax_aborted
	xmax_lock_only
	keys_updated
(0,1)	ctid
мама мыла раму	данные





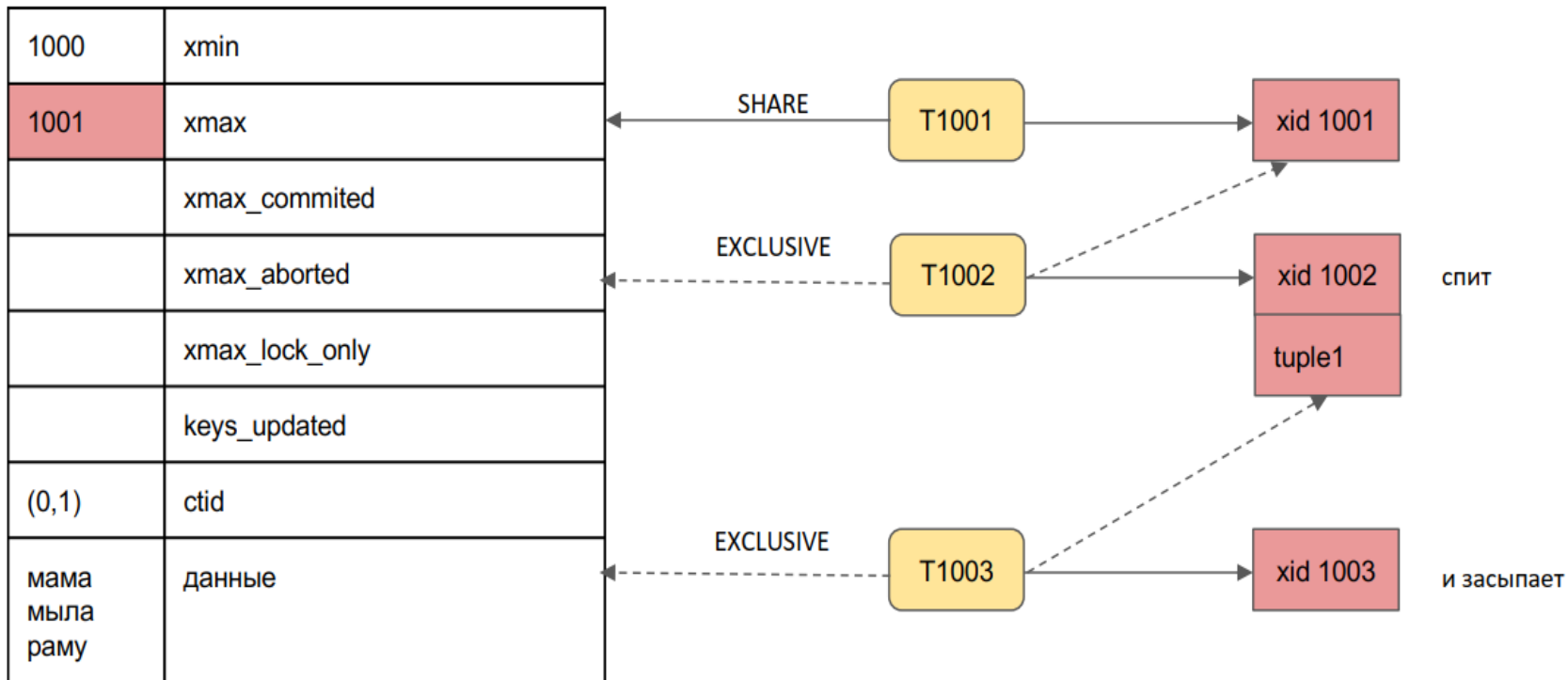
# Блокировки строк

Как это происходит



# Блокировки строк

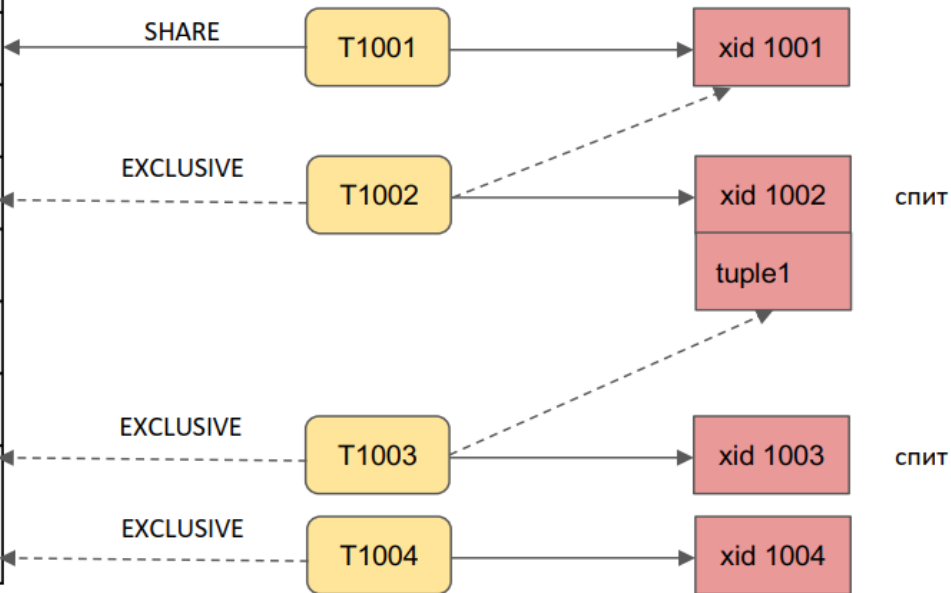
Как это происходит



## Блокировки строк

## Как это происходит

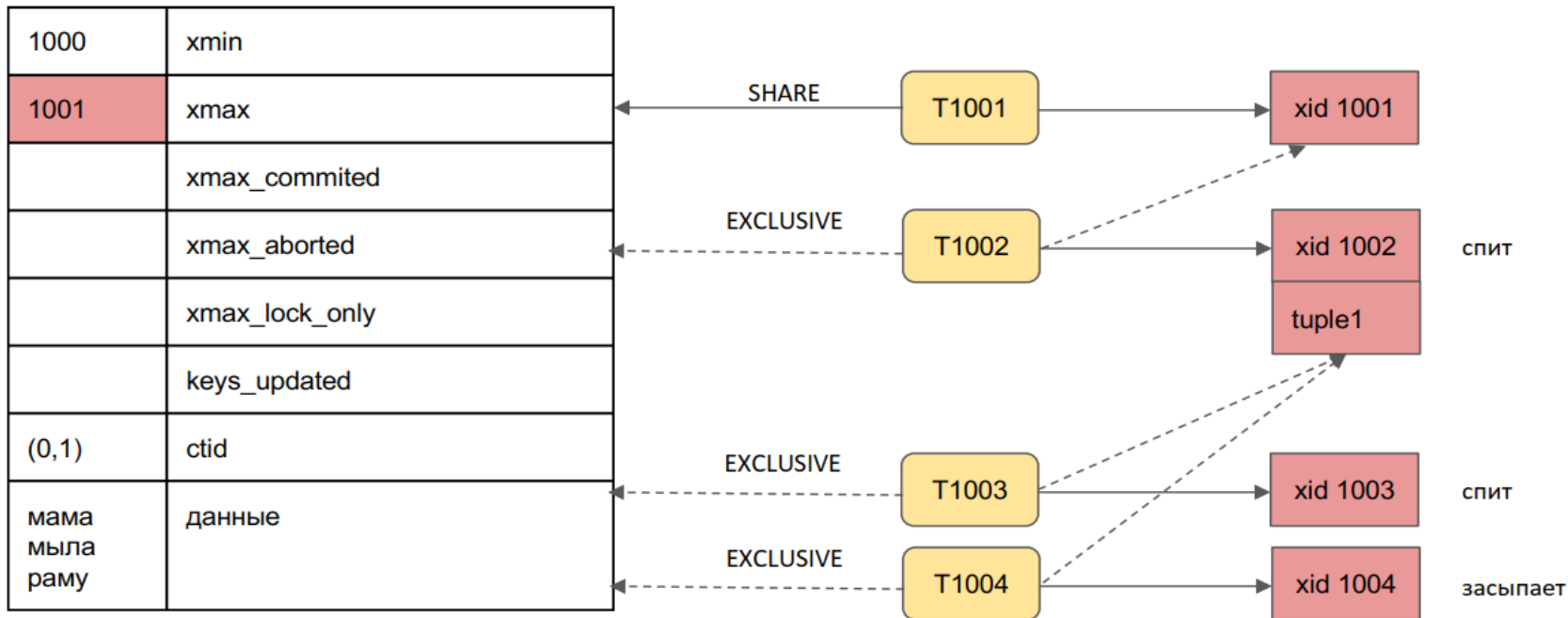
1000	xmin
1001	xmax
	xmax_committed
	xmax_aborted
	xmax_lock_only
	keys_updated
(0,1)	ctid
мама мыла раму	данные



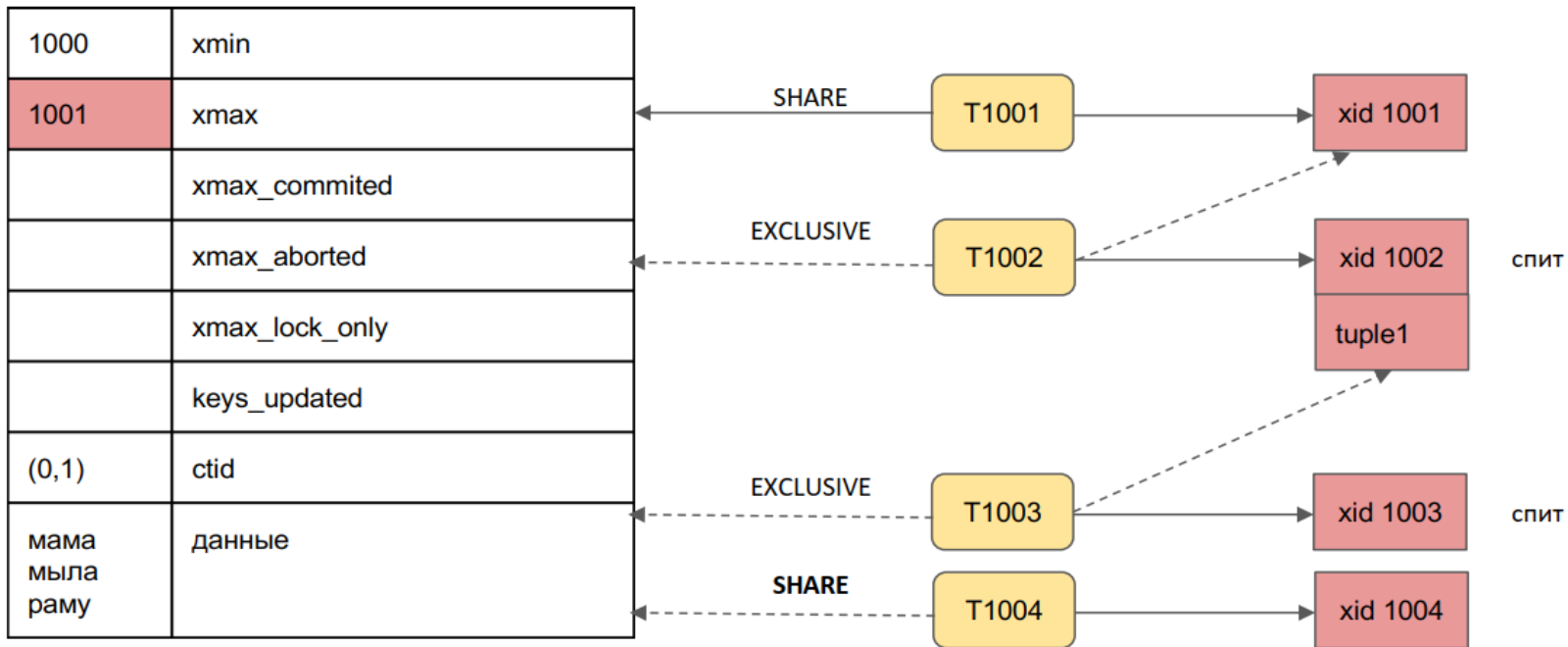
что произойдет дальше?

# Блокировки строк

Как это происходит



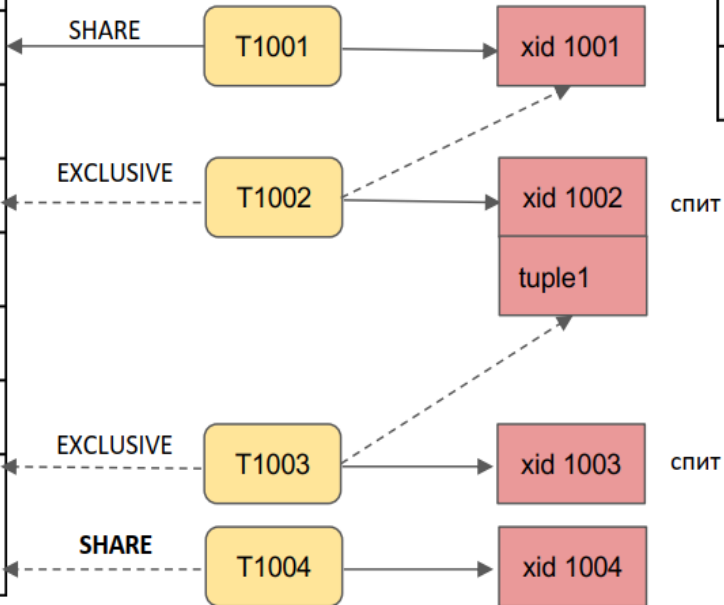
# Блокировки строк



что произойдет дальше?

# Блокировки строк

1000	xmin
2	xmax
	xmax_committed
	xmax_aborted
	xmax_lock_only
t	xmas_is_multi
(0,1)	ctid
мама мыла раму	данные

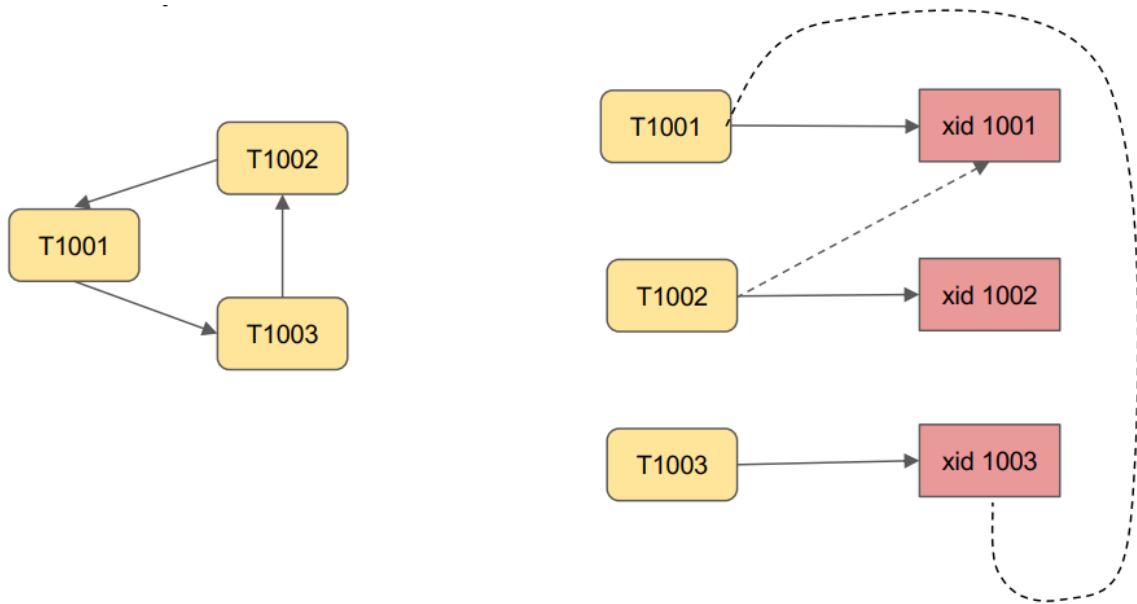


\$PGDATA/pg\_multixact

1001	SHARE
1004	SHARE

# Блокировки строк

## Взаимоблокировки



postgres строит граф связей для выявления таких ситуаций  
deadlock\_timeout

# Блокировки в памяти



# Блокировки в памяти. Spin блокировки

## Spinlock

- устанавливаются на очень короткое время, несколько инструкций процессора
- используются атомарные инструкции процессора
- единственный режим — исключительный
- нет возможности мониторинга
- нет обнаружения взаимоблокировок
- цикл активного ожидания

# Блокировки в памяти. Легкие блокировки

## LightWeightLock

- устанавливаются на короткое время, обычно доли секунды
- исключительный и разделяемый режимы
- есть мониторинг
- нет обнаружения взаимоблокировок
- пассивное ожидание
- при освобождении ресурса возникает состояние гонки, выигрывает случайный процесс

# Блокировки в памяти. Закрепление буфера

## BufferPin

- устанавливается на время работы с буфером, возможно длительное
- исключительный и разделяемый режимы
- есть мониторинг
- есть обнаружение взаимоблокировок
- пассивное ожидание, но обычно закрепленный буфер пропускается

# Блокировки в памяти. Мониторинг

Когда процесс ожидает чего-либо, этот факт отражается в представлении

**pg\_stat\_activity:**

- wait\_event\_type — тип ожидания
- wait\_event — имя конкретного ожидания
- информация может быть не полна
- охвачены не все места в коде, в которых могут быть ожидания
- информация только на текущий момент
- единственный способ получить картину во времени — семплинг
- достоверная картина только при большом числе измерений

# Вывод в файл медленных запросов

`log_min_duration_statement` (integer) в миллисекундах

# ДЗ



# Домашнее задание

1. Настройте сервер так, чтобы в журнал сообщений сбрасывалась информация о блокировках, удерживаемых более 200 миллисекунд. Воспроизведите ситуацию, при которой в журнале появятся такие сообщения.

2. Смоделируйте ситуацию обновления одной и той же строки тремя командами UPDATE в разных сеансах. Изучите возникшие блокировки в представлении pg\_locks и убедитесь, что все они понятны. Пришлите список блокировок и объясните, что значит каждая.

3. Воспроизведите взаимоблокировку трех транзакций. Можно ли разобраться в ситуации постфактум, изучая журнал сообщений?

4. Могут ли две транзакции, выполняющие единственную команду UPDATE одной и той же таблицы (без where), заблокировать друг друга?

\* Попробуйте воспроизвести такую ситуацию.

# Рефлексия



# Рефлексия



С какими впечатлениями уходите с вебинара?



**Заполните, пожалуйста,  
опрос о занятии  
по ссылке в чате**

Спасибо за внимание!

# Приходите на следующие вебинары



**Коробков Виктор**