

БЛОК 3. АРХИТЕКТУРА

БЛОКИРОВКИ

A white icon of a pair of curly braces "{ }", commonly used to represent code or programming.

{ }

A white icon of the word "begin" in a lowercase, sans-serif font, positioned inside a blue square.

begin



ЦЕЛЬ УРОКА



1

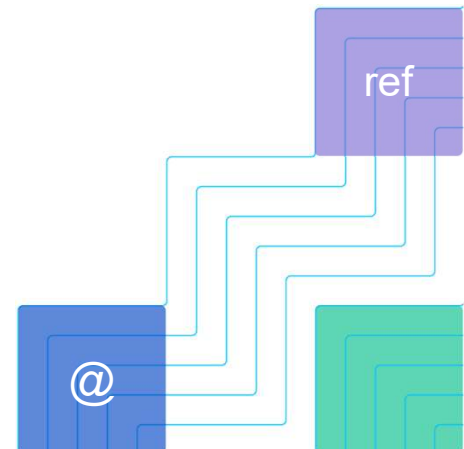
Узнать, что такое
блокировки

2

Понять какие виды
существуют

3

Узнать, что такое
дедлоки



СОДЕРЖАНИЕ УРОКА



1

Блокировки

2

Виды блокировок

3

Дедлоки



ref

@



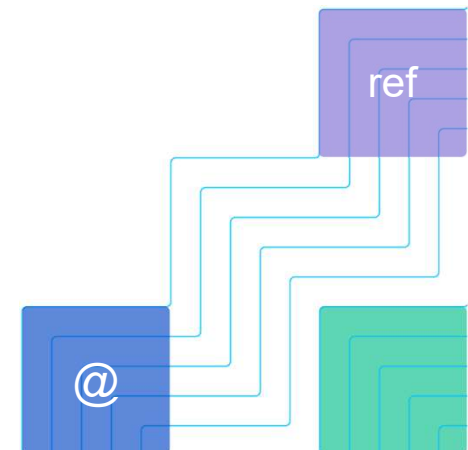
БЛОКИРОВКИ ОБЪЕКТОВ



Проблема: как одновременно получить доступ к одному объекту?

Решение: упорядочение конкурентного доступа к разделяемым ресурсам

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





БЛОКИРОВКИ ОБЪЕКТОВ



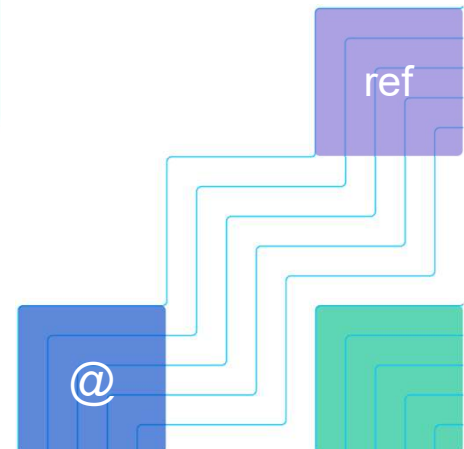
Дальнейшее развитие

МНОГОВЕРСИОННОСТЬ

несколько версий данных, не избавляет от всех проблем

ОПТИМИСТИЧНЫЕ БЛОКИРОВКИ

процессы не блокируются, но при неудачном стечении обстоятельств возникает ошибка



БЛОКИРОВКИ ОБЪЕКТОВ. ВИДЫ РЕСУРСОВ

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. ВИДЫ БЛОКИРОВОК



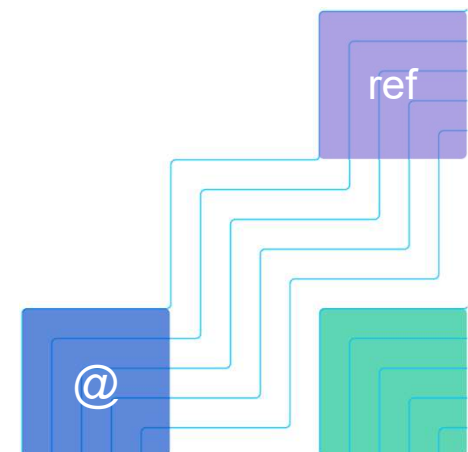
По типу:

EXCLUSIVE

Исключительные

SHARED

Совместные





БЛОКИРОВКИ ОБЪЕКТОВ. ВИДЫ БЛОКИРОВОК



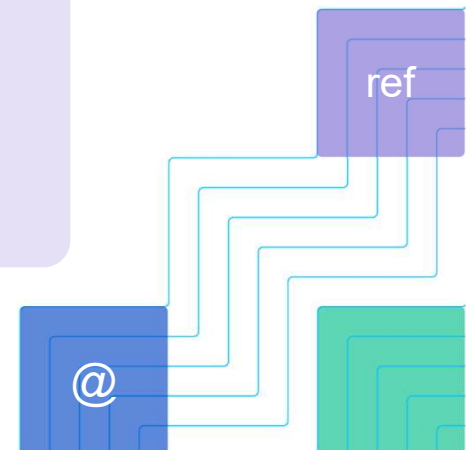
По времени жизни:

ДОЛГОВРЕМЕННЫЕ БЛОКИРОВКИ

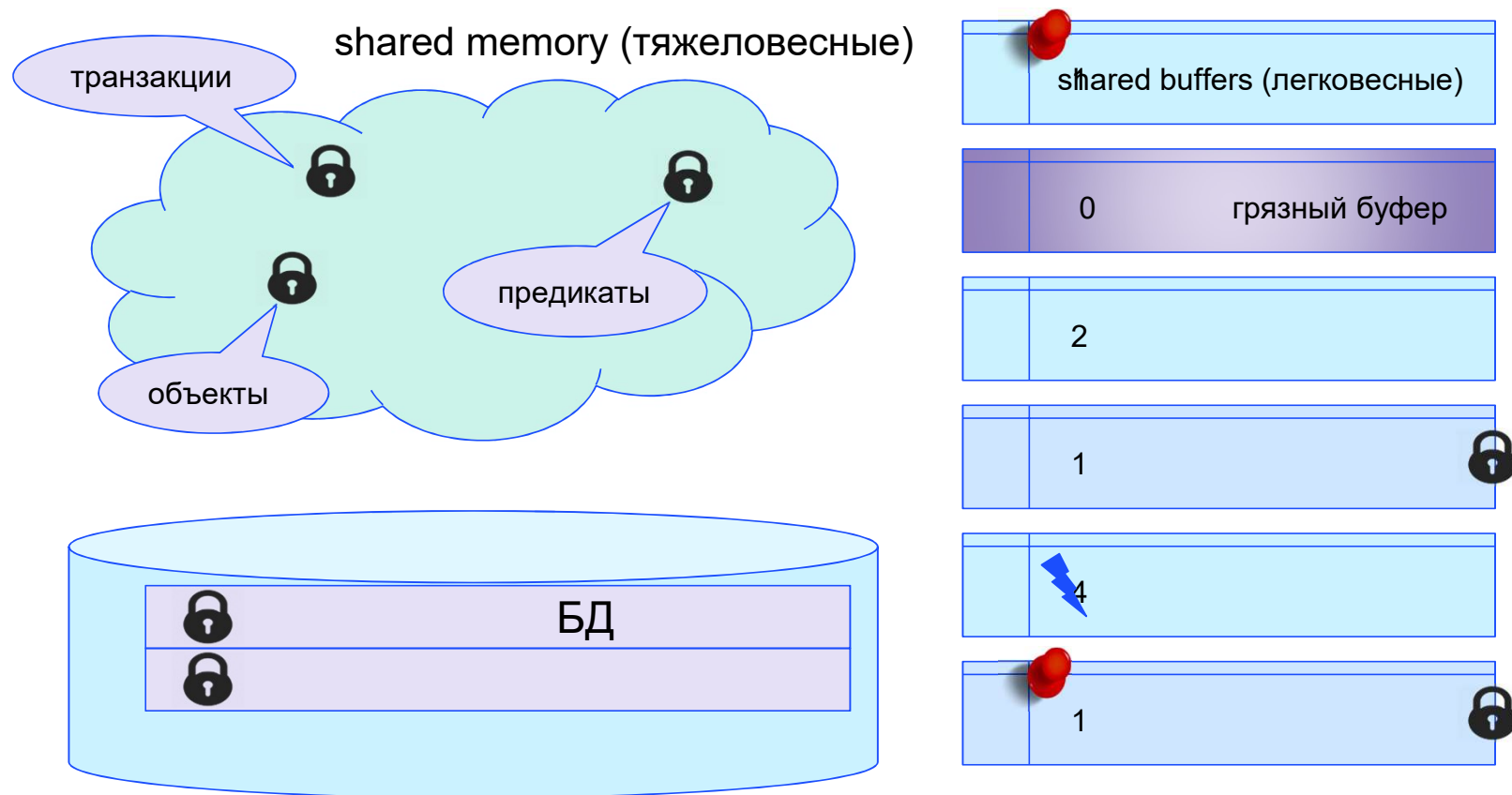
- обычно захватываются до конца транзакции и относятся к хранимым данным (например таблицы и отношения)
- большое число режимов
- развитая «тяжеловесная» инфраструктура, мониторинг

SHARED

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



БЛОКИРОВКИ ОБЪЕКТОВ. КАКИЕ СУЩЕСТВУЮТ



1. БЛОКИРОВКИ ОТНОШЕНИЙ



тип ресурса: relation в pg_locks

Режимы:

- Access Share
- Row Share
- Row Exclusive
- Share Update Exclusive
- Share
- Share Row Exclusive
- Exclusive
- **CONCURRENTLY**
- Access Exclusive

SELECT
SELECT FOR UPDATE/SHARE
UPDATE, DELETE, INSERT
VACUUM, ALTER TABLE,
CREATE INDEX CONCURRENTLY
CREATE INDEX
CREATE TRIGGER, ALTER TABLE
REFRESH MAT. VIEW

DROP, TRUNCATE, VACUUM FULL,
LOCK TABLE, ALTER TABLE, REFRESH MAT. VIEW

допускают
изменение
данных

@

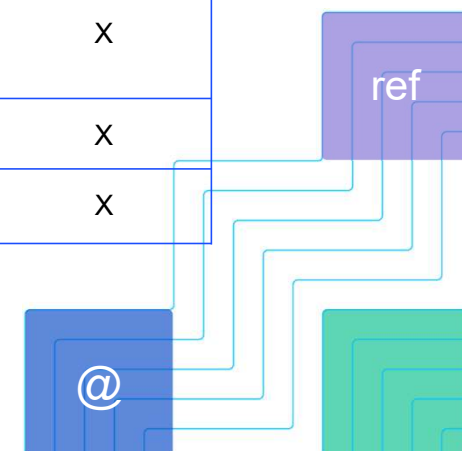
ref

1. БЛОКИРОВКИ ОТНОШЕНИЙ.

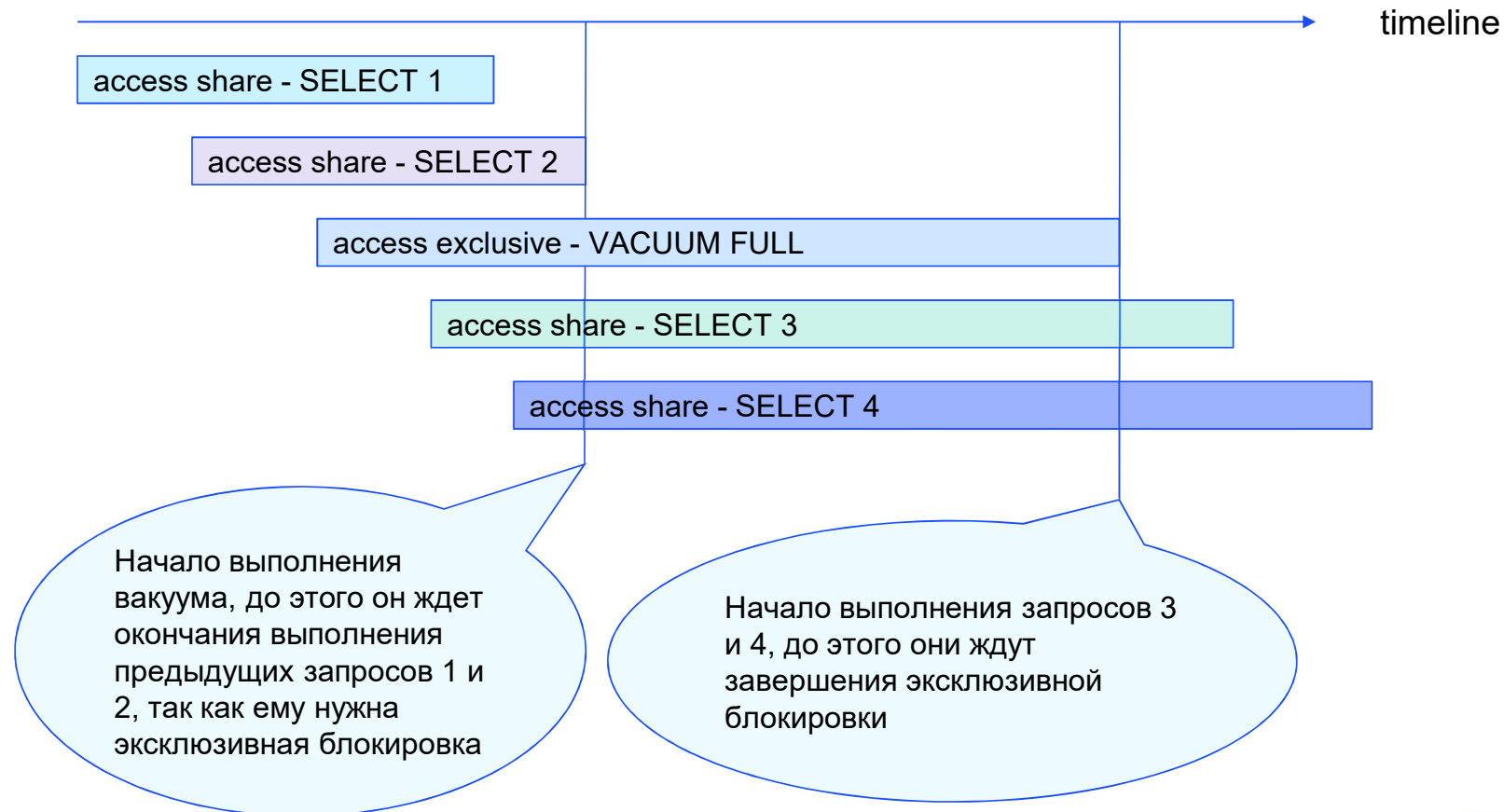
X – обозначены не совместимые блокировки.



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



1. БЛОКИРОВКИ ОТНОШЕНИЙ

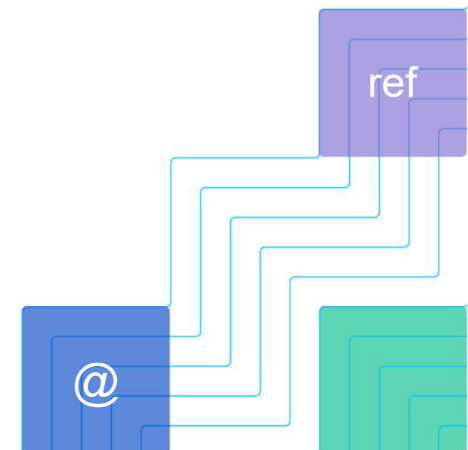




2. БЛОКИРОВКИ СТРОК



- информация только в страницах данных [на диске](#)
- в отличие от блокировок отношений (которых ограниченное количество), здесь ограничений нет (могут быть миллиарды строк)
- поле xтах заголовка версии строки + информационные биты
- большое число минимально отражается на производительности
- очередь ожидания организована с помощью блокировок объектов (пропорционально числу процессов, а не строк)



2. БЛОКИРОВКИ СТРОК. РЕЖИМЫ

Эксклюзивный и разделяемый. Эксклюзивный:

1 **UPDATE:** удаление строки или изменение всех полей

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

2 **NO KEY UPDATE:** изменение любых полей, кроме ключевых

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

2. БЛОКИРОВКИ СТРОК. РЕЖИМЫ



Разделяемый:

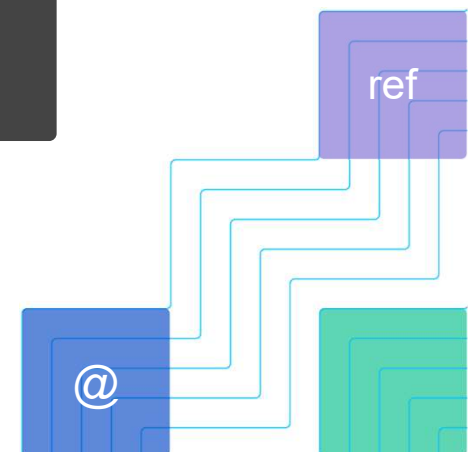
- 1 **SHARE:** запрет изменения любых полей строки`

```
SELECT FOR SHARE
```

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

```
SELECT FOR KEY SHARE
```

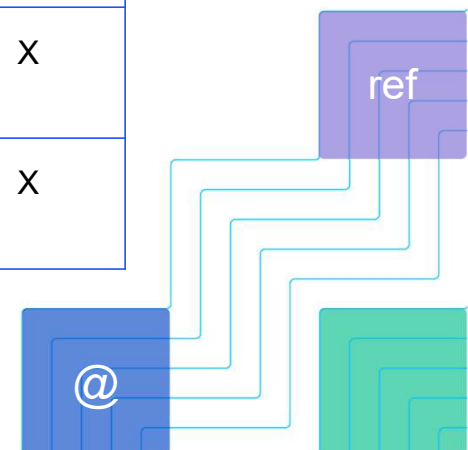
```
UPDATE (без изменения ключевых полей)
```



2. БЛОКИРОВКИ СТРОК. РЕЖИМЫ

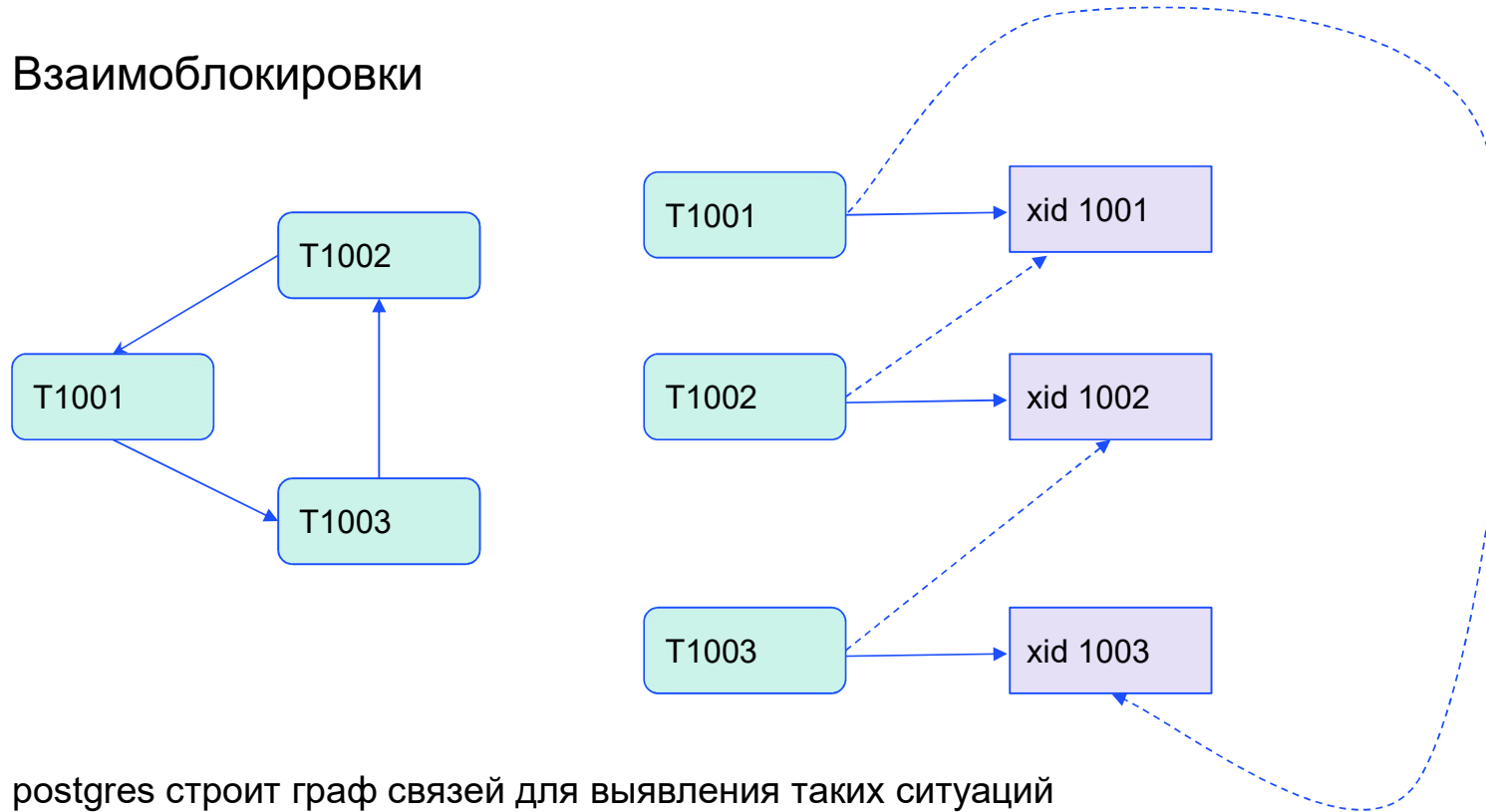


Запрашиваемый режим блокировки	Текущий режим блокировки			
	FOR KEY SHARE	FOR SHARE	FOR NO KEY UPDATE	FOR UPDATE
FOR KEY SHARE				X
FOR SHARE			X	X
FOR NO KEY UPDATE		X	X	X
FOR UPDATE	X	X	X	X



2. БЛОКИРОВКИ СТРОК. ДЕДЛОК

Взаимоблокировки



postgres строит граф связей для выявления таких ситуаций
deadlock_timeout

2. БЛОКИРОВКИ СТРОК. ПРАКТИКА



Как выглядит взаимоблокировка

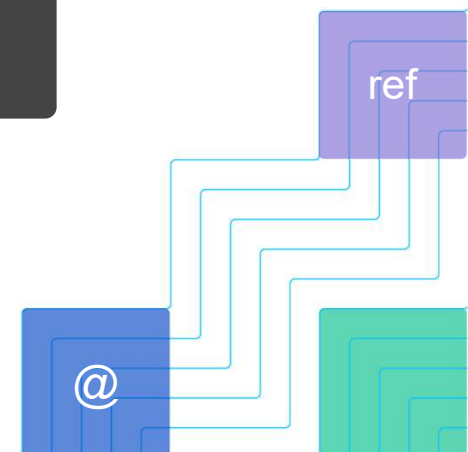
```
ERROR: deadlock detected
```

```
DETAIL: Process 1092 waits for ShareLock on transaction 32001;  
blocked by process 1093.
```

```
Process 1093 waits for ShareLock on transaction 32002; blocked by  
process 1092.
```

```
HINT: See server log for query details.
```

```
CONTEXT: while updating tuple (0,2) in relation "accounts"
```



3. БЛОКИРОВКИ В ПАМЯТИ. SPINLOCK



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

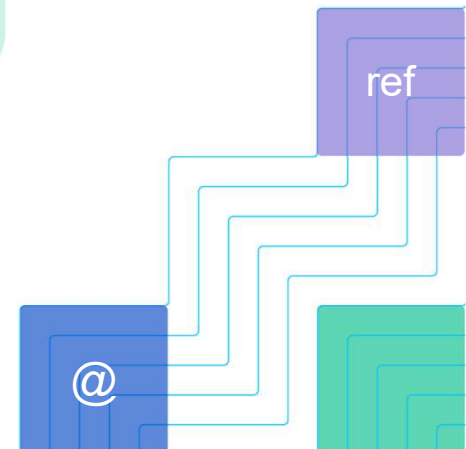
3. БЛОКИРОВКИ В ПАМЯТИ. ЛЕГКИЕ БЛОКИРОВКИ



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





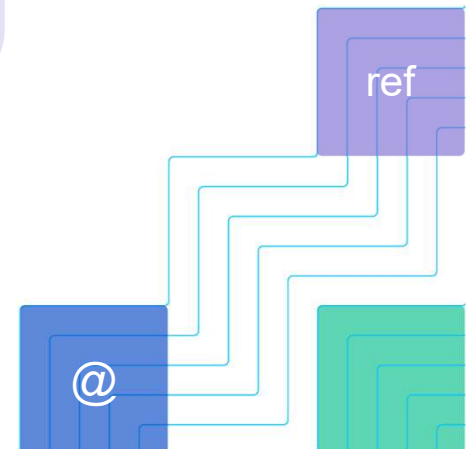
3. БЛОКИРОВКИ В ПАМЯТИ. ЗАКРЕПЛЕНИЕ БУФЕРА



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





ИТОГИ ЗАНЯТИЯ



01



Узнали, что такое
блокировки

02

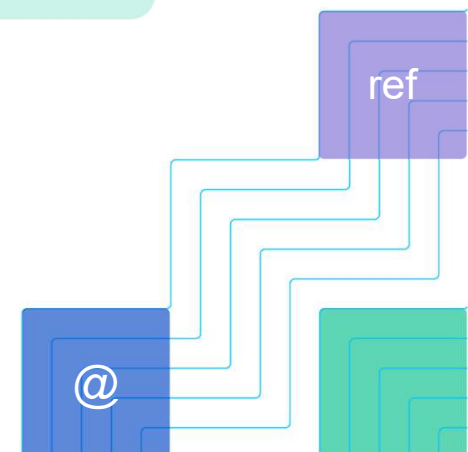


Поняли какие виды
существуют

03



Узнали, что такое
дедлоки



ПРОВЕРОЧНОЕ ЗАДАНИЕ

Цель задания: найти ситуации мертвых блокировок

{ }

begin



ЗАДАНИЕ ДЛЯ САМОПРОВЕРКИ



01

использовать
существующую или
создать новую
таблицу с данными

02

в 2 сеансах
начать
транзакции

03

в 1 сеансе сделать
обновление первой
строки таблицы

04

во 2 сеансе сделать
обновление второй
строки таблицы

05

в 1 сеансе сделать
обновление второй
строки таблицы

06

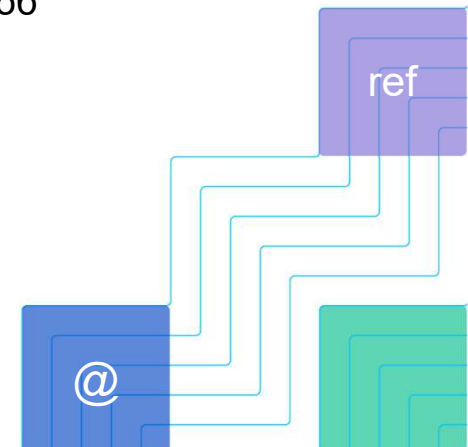
во 2 сеансе
сделать
обновление первой
строки таблицы

07

убедиться, что
произошел дедлок

08

в лог файле найти
информацию об
этом



СПАСИБО

На следующем занятии мы переходим к следующему блоку Организация данных:

- Логическое устройство. Базы данных и схемы
- Табличные пространства
- Физическое устройство

{ }

begin