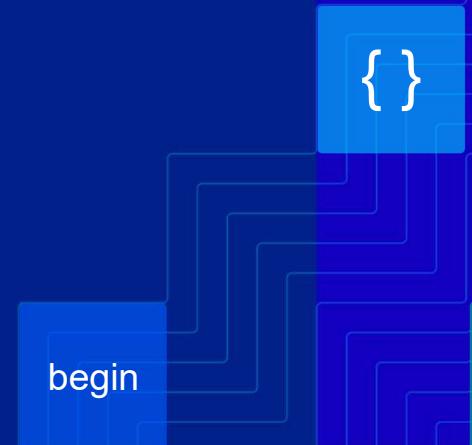


БЛОК 3. АРХИТЕКТУРА

ВАКУУМ И АВТОВАКУУМ



begin

{ }

ЦЕЛЬ УРОКА

01

Понять, что такое
вакуум и
автовакуум

02

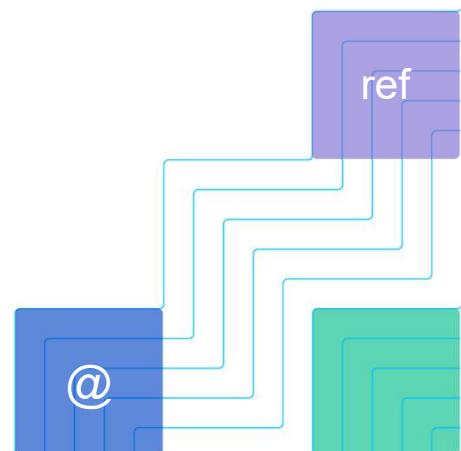
Определить их задачи,
архитектуру и их практическое
применение

03

Узнать как поменять
настройки этих процессов

04

Понять, что такое
заморозка и на что она
влияет



СОДЕРЖАНИЕ УРОКА

1

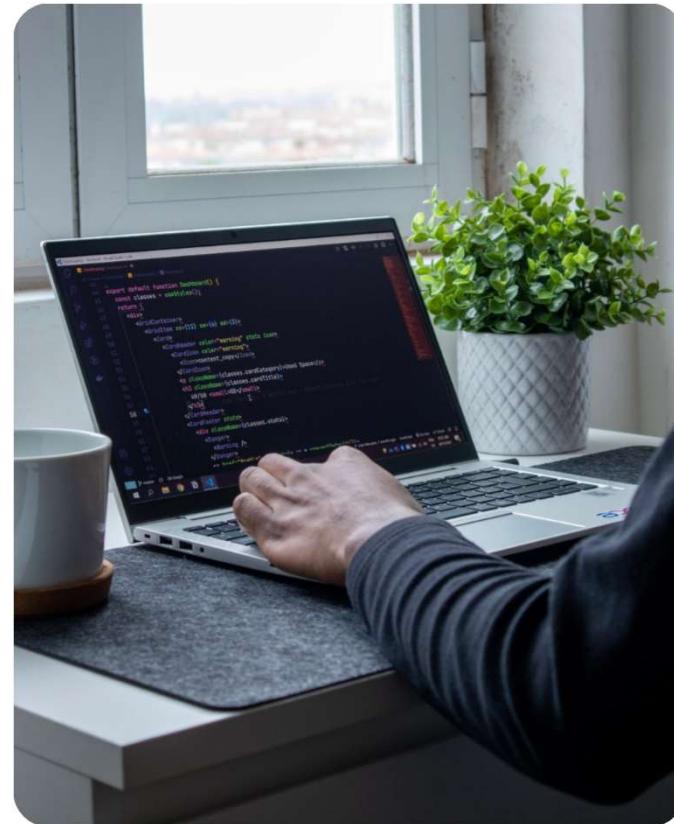
Вакуум

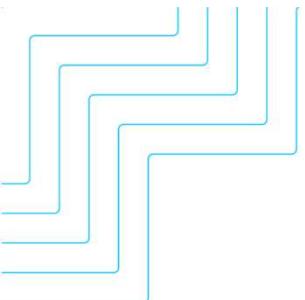
2

Автовакуум

3

Настройки





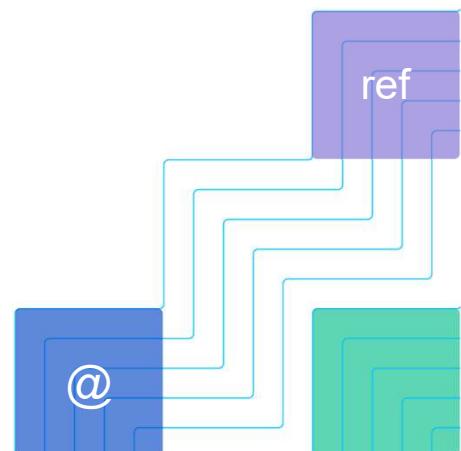
ВАКУУМ

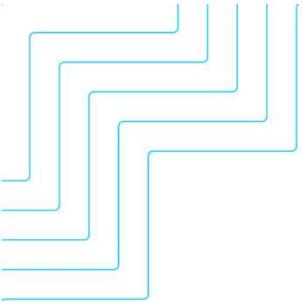


Неиспользуемые в рамках незавершенных транзакций строки называются мертвыми (dead)

Они, равно как и соответствующие им записи в индексах, будут присутствовать в БД вплоть до принудительного удаления — т.н. vacuum

Ну и пусть?



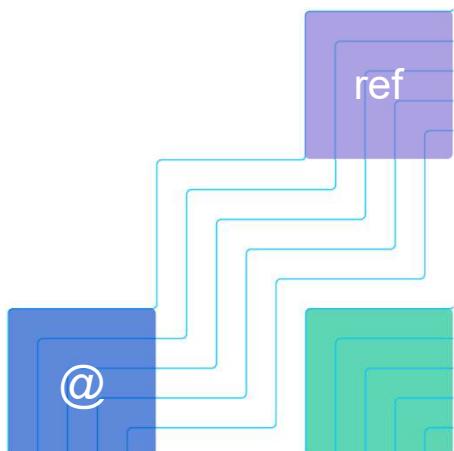


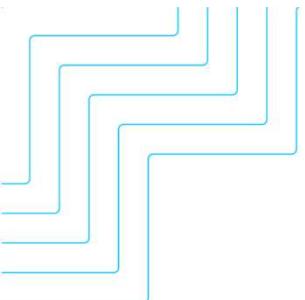
ВАКУУМ



Но:

- ✗ занимают место в памяти
- ✗ занимают место на дисках
- ✗ участвуют в select и update, where
- ✗ достаточно ощутимо и негативно влияют на производительность





ВАКУУМ



Варианты использования:

VACUUM

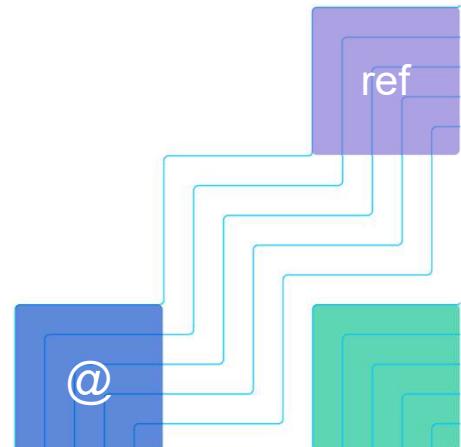
VACUUM VERBOSE

VACUUM ANALYZE

Выполняет очистку
(VACUUM), а затем
анализ (ANALYZE) всех
указанных таблиц

VACUUM FULL

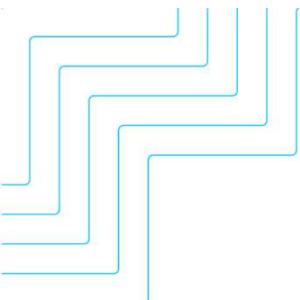
Дефрагментация. нужен
исключительный доступ
без блокировок на
запись. Создается новый
файл!!!



За выполнением можно следить через:

- 1 консоль в случае vacuum verbose
- 2 или через системное представление

```
SELECT * FROM pg_stat_progress_vacuum;
```



ВАКУУМ. ПРАКТИКА



Сценарий —

01

В 1 окне обновим
запись

02

Во 2 окне делаем begin
transaction isolation level
repeatable read

03

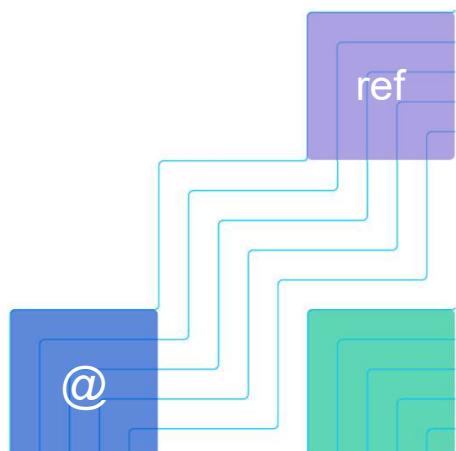
В 1 пообновляем
еще

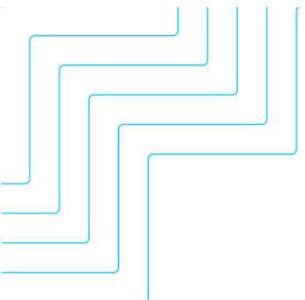
04

Смотрим vacuum verbose

Для дополнительного чтения:

[MVCC-6. Очистка / Блог компании Postgres Professional / Хабр](#)



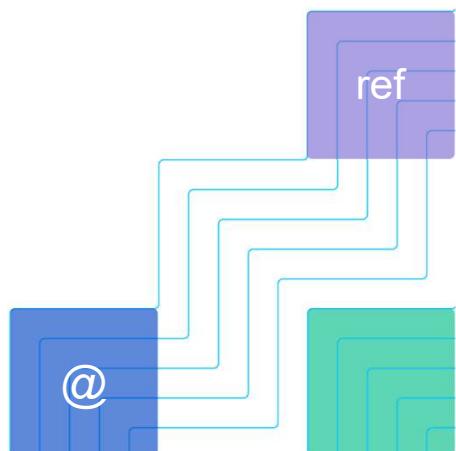


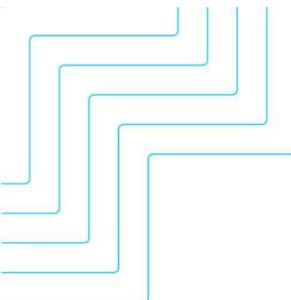
АВТОВАКУУМ



Чтобы вручную не заниматься очисткой таблиц есть фоновый процесс [Автовакуум](#)

Он [не работает все время](#), а вызывает вакуум в зависимости от различных порогов срабатывания





АВТОВАКУУМ. НАСТРОЙКИ

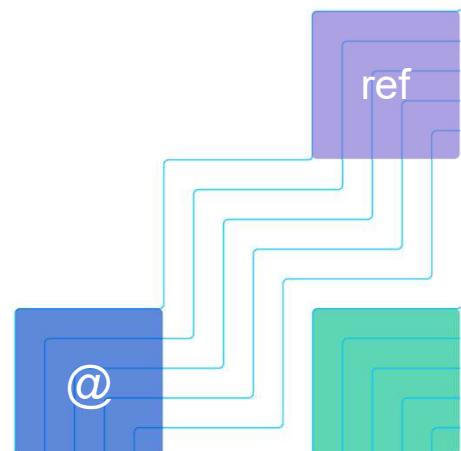


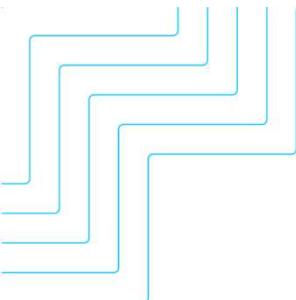
Желательно настраивать максимально агрессивно:

`autovacuum_max_workers = 10` — количество параллельных процессов

`autovacuum_naptime = 15s` — время сна между запусками процессов автовакуума

`autovacuum_vacuum_threshold = 25` — минимальное количество измененных строк, при котором автовакуум придет в таблицу для очистки





АВТОВАКУУМ. НАСТРОЙКИ



Но не забывать что autovacuum может стать источником проблем:

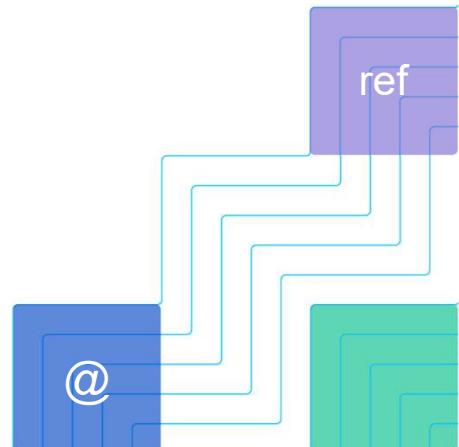
- например высокий ввод-вывод

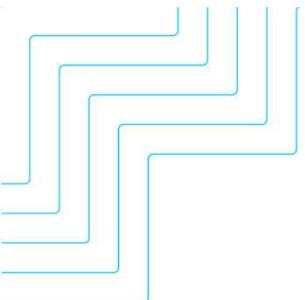
<https://postgrespro.ru/docs/postgresql/14/runtime-config-autovacuum>
[MVCC-7. Автоочистка / Блог компании Postgres Professional / Хабр](#)



Как амазон решил проблему с автовакуумом:

<https://aws.amazon.com/ru/blogs/database/a-case-study-of-tuning-autovacuum-in-amazon-rds-for-postgresql/>





АВТОВАКУУМ. НАСТРОЙКИ



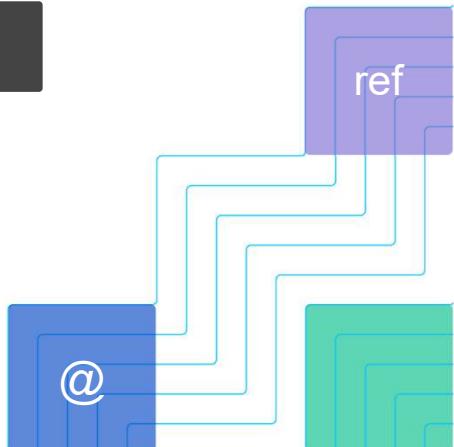
Посмотрим на настройки:

```
select name, setting, context, short_desc from pg_settings where category like  
'%Autovacuum%';
```

```
ALTER SYSTEM SET autovacuum_naptime = 55;  
SHOW autovacuum_naptime;
```

Мягкий релоад конфигурации без рестарта сервера:

```
pg_ctlcluster 14 main reload  
SELECT pg_reload_conf();
```



ЗАМОРОЗКА. ПРОБЛЕМА

transaction id wraparound

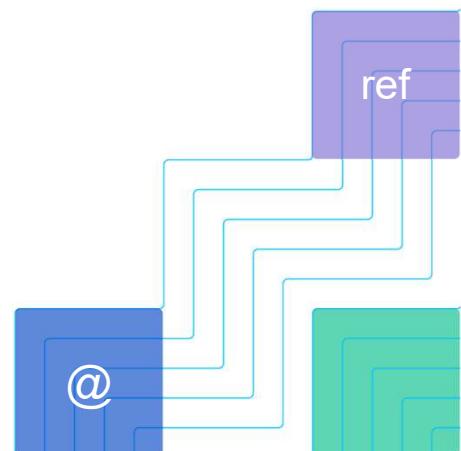
Под номер транзакции в PostgreSQL выделено 32 бита (около 4 млрд)

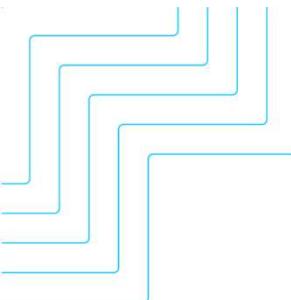
При активной работе сервера - нагрузке 1000 транзакций в секунду
переполнение произойдет всего через [полтора месяца непрерывной работы](#).

Механизм многоверсионности полагается на последовательность нумерации — тогда из двух транзакций [транзакцию с меньшим номером можно считать начавшейся раньше](#).

Поэтому нельзя просто так [обнулить](#) счетчик и продолжить нумерацию [заново](#).

Поэтому придумали механизм заморозки.

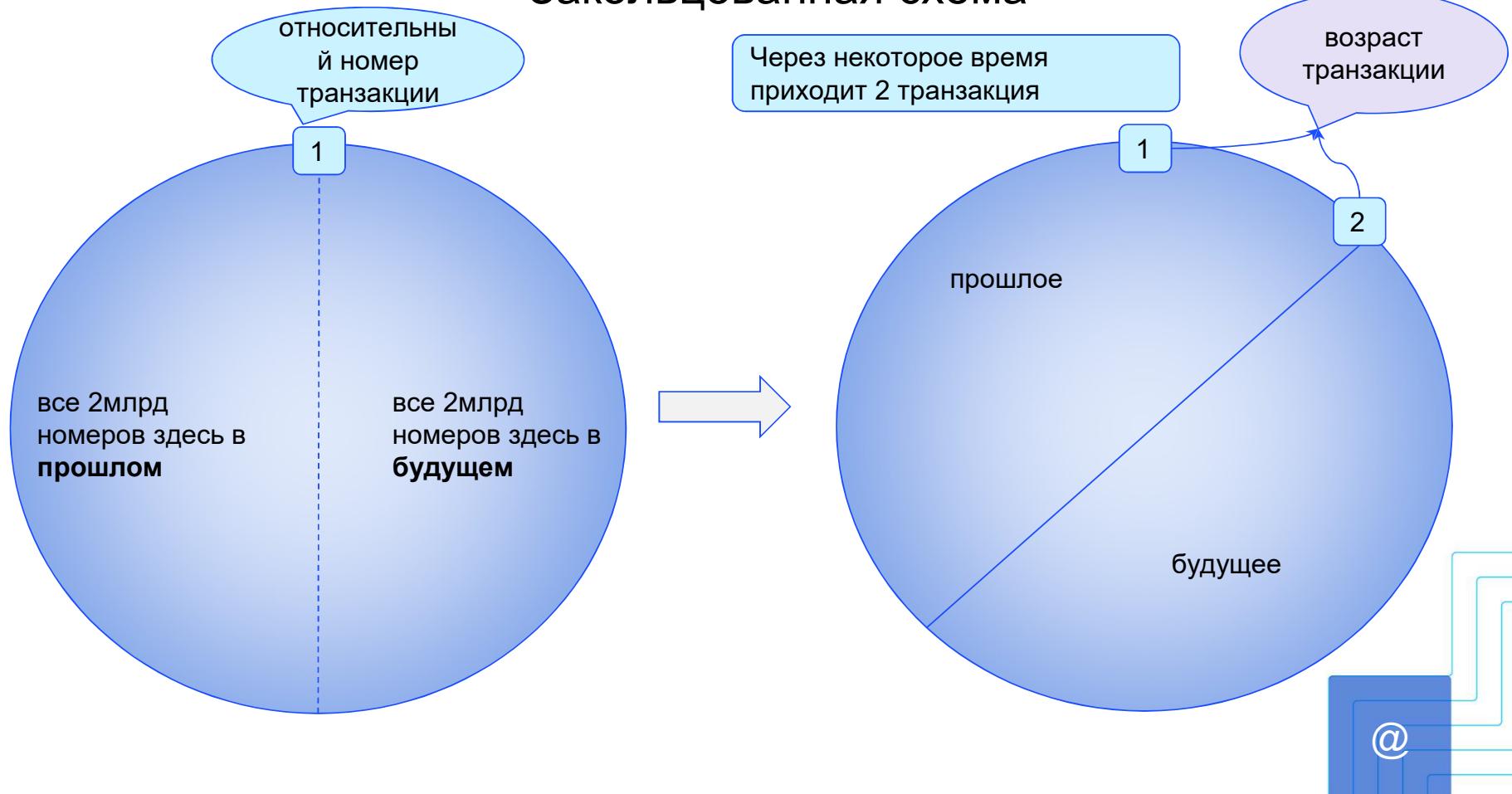




ЗАМОРОЗКА



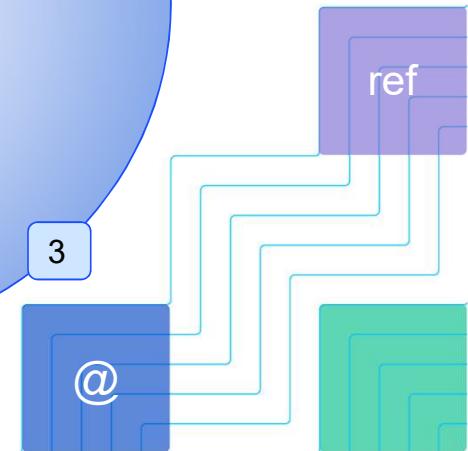
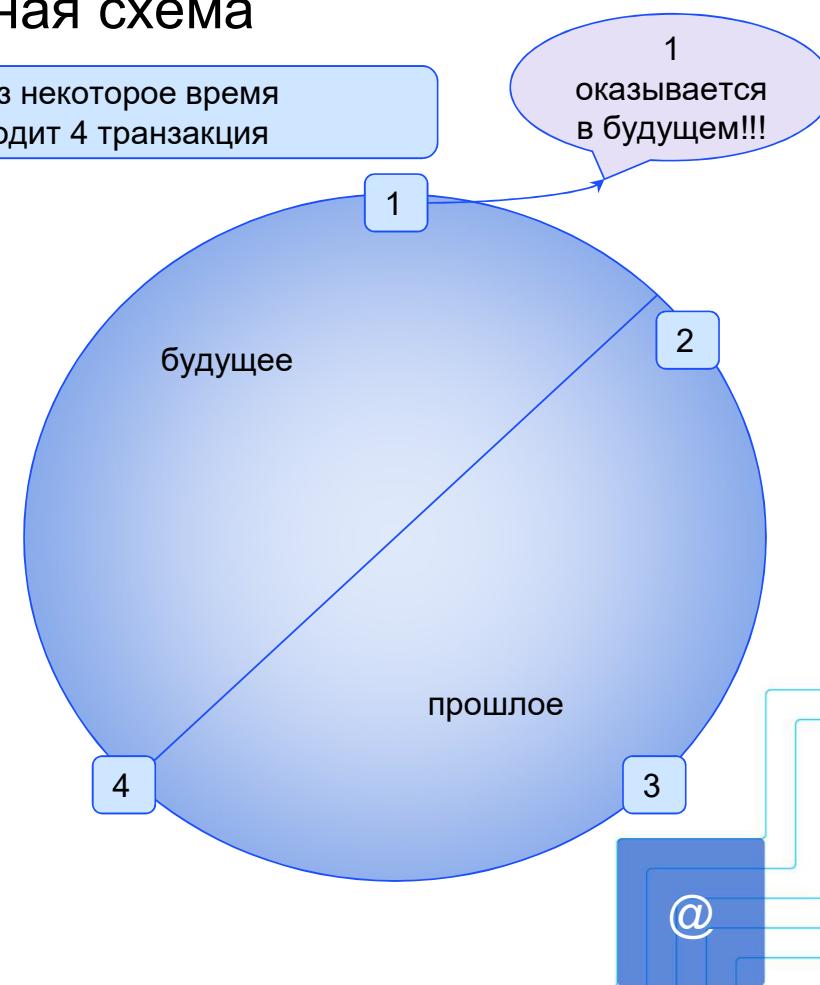
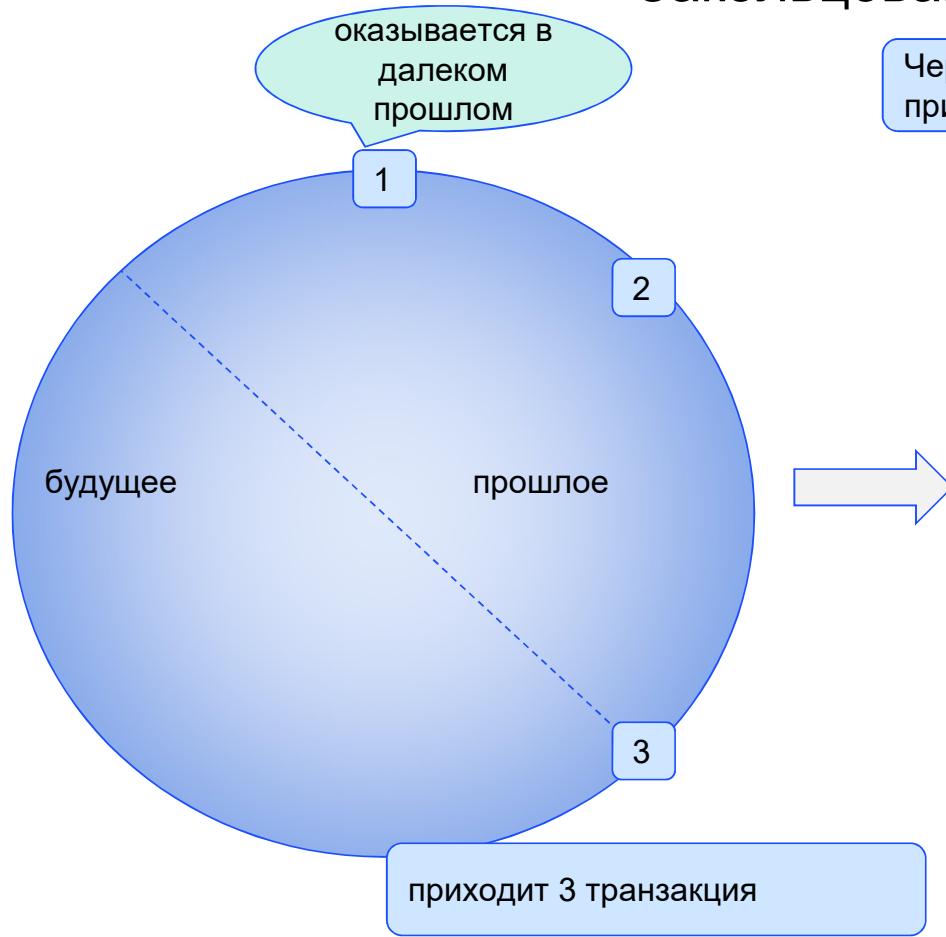
Закольцованная схема

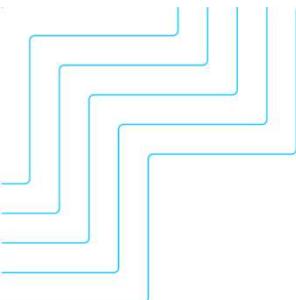


ЗАМОРОЗКА



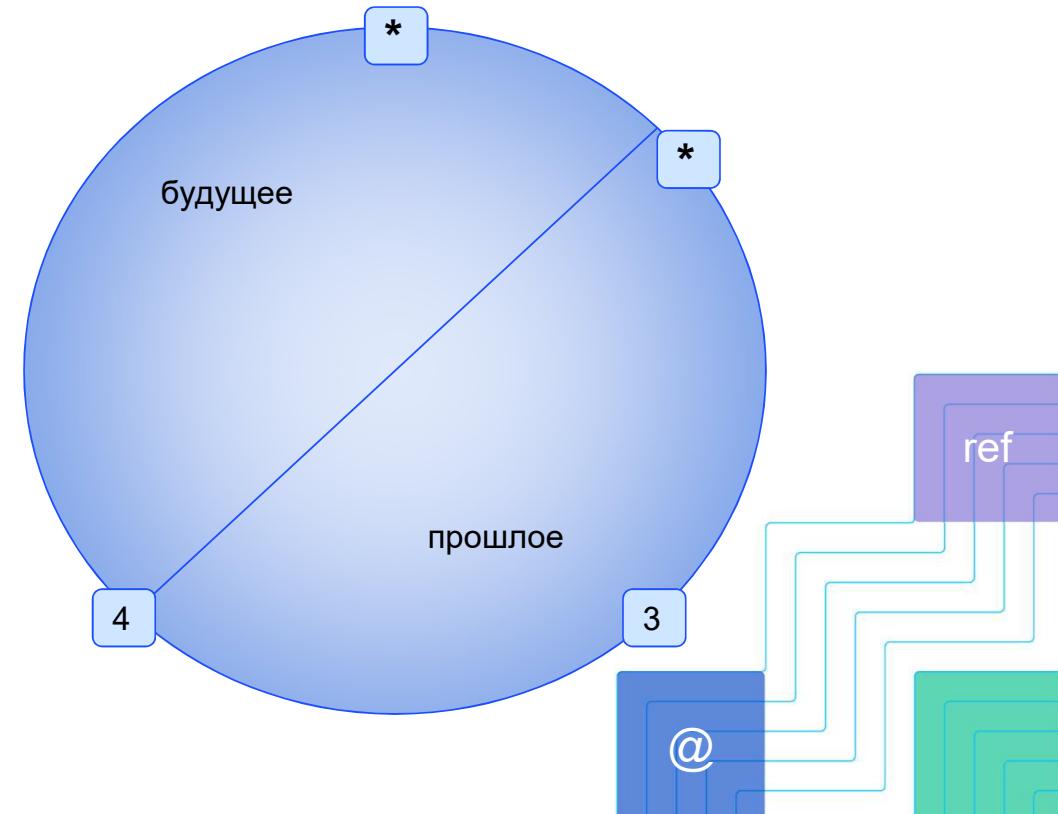
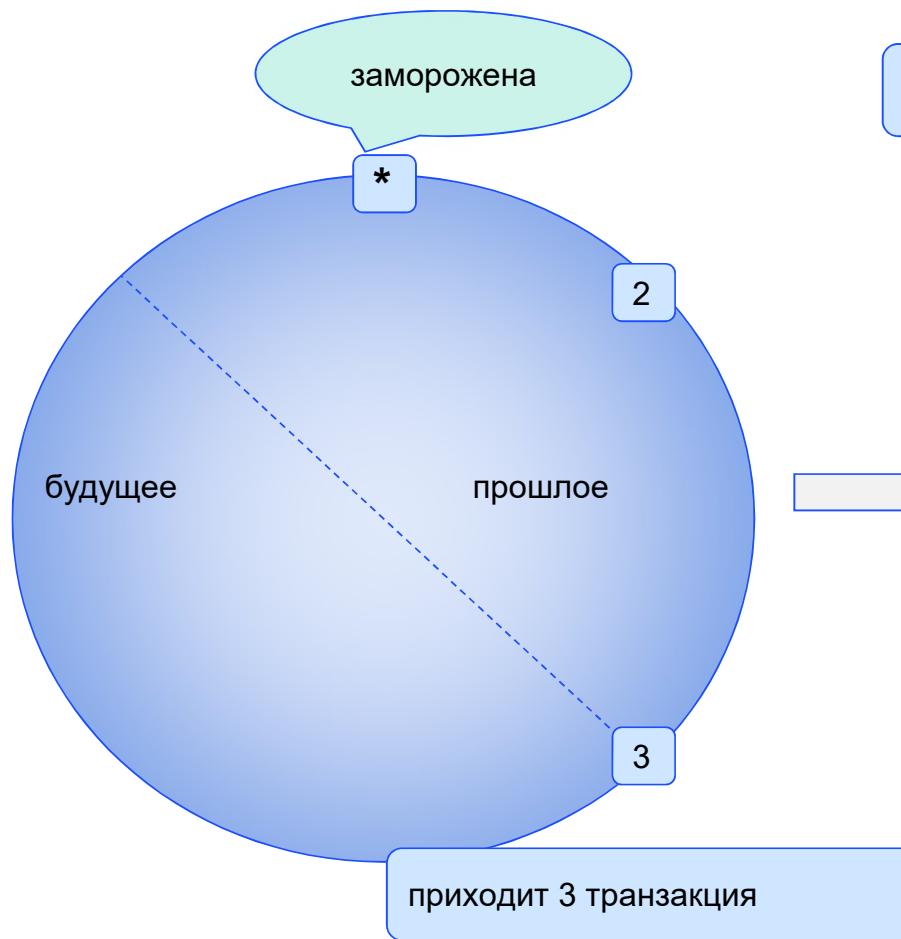
Закольцованная схема

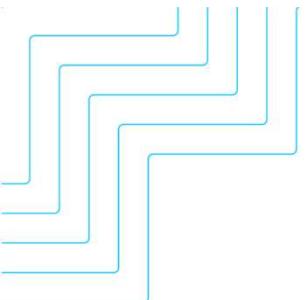




ЗАМОРОЗКА

Закольцованная схема. Проблема решена

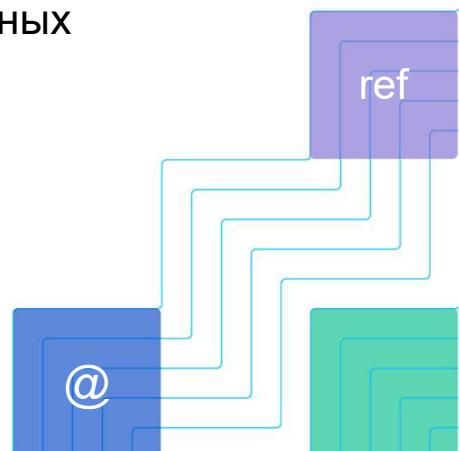




ЗАМОРОЗКА



- 1 Для того, чтобы пометить номер транзакции `xmin` как замороженный, выставляются одновременно оба бита-подсказки — бит фиксации и бит отмены
- 2 Одна из задач автовакуума пробегать и замораживать старые строки
- 3 Замороженная версия строки считается старше любых обычных данных и всегда видна во всех снимках данных



ЗАМОРОЗКА. НАСТРОЙКА

3 параметра настройки:

```
SHOW vacuum_freeze_min_age;
```

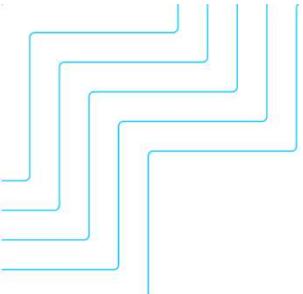
Плохо делать очень маленьким для активно меняющихся данных — двойная работа автовакуума:

```
SHOW vacuum_freeze_table_age;
```

Вакуум пробегает полностью заполненные и неменяющиеся странички, куда обычно не заходит

```
SHOW autovacuum_freeze_max_age;
```

для аварийной заморозки, иначе:



ИТОГИ ЗАНЯТИЯ



01



Поняли, что такое
вакуум и автовакуум

02



Поняли их задачи, архитектуру
и их практическое применение

04

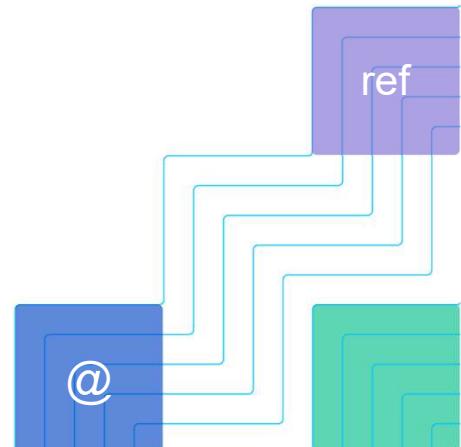


Узнали как поменять
настройки этих процессов

03

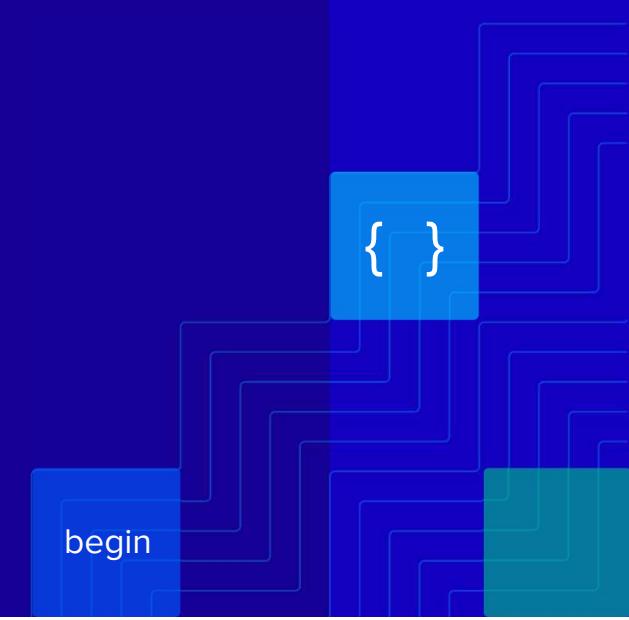


Поняли, что такое
заморозка и на что она
влияет



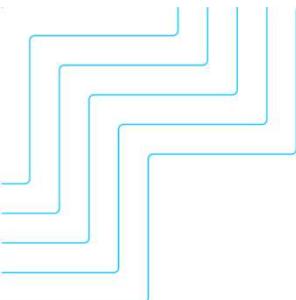
ПРОВЕРОЧНОЕ ЗАДАНИЕ

Цель задания: понять механизм появления мертвых строк и процесс их утилизации



The diagram illustrates the deallocation of memory blocks. It features a vertical stack of blue rectangles of decreasing size from top to bottom. The top rectangle contains the symbol '{' and the bottom one contains the word 'begin'. A green rectangle is positioned at the bottom right. Light blue wavy lines connect the top of the first rectangle to the bottom of the second, and the bottom of the second to the top of the third, representing the allocation and deallocation of memory blocks.

begin



ЗАДАНИЕ НА САМОПРОВЕРКУ



01

Убедиться, что
автокоммит включен

02

На предыдущем задании для
самопроверки была создана
таблица `persons`, можно
использовать её или создать
новую

03

Обновить 1 запись в
этой таблице

04

Убедится, что появилась
мертвая запись

05

Очистить таблицу от
старых записей вызвав
вакуум

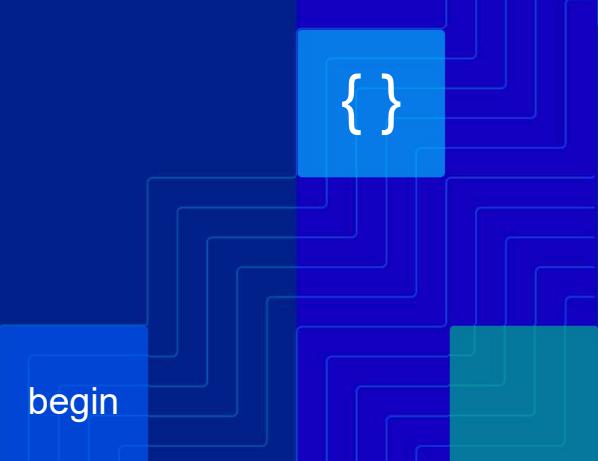
06

Убедится, что остались
только актуальные
записи



СПАСИБО

На следующем занятии мы рассмотрим тему Буферного кеша более подробно



begin