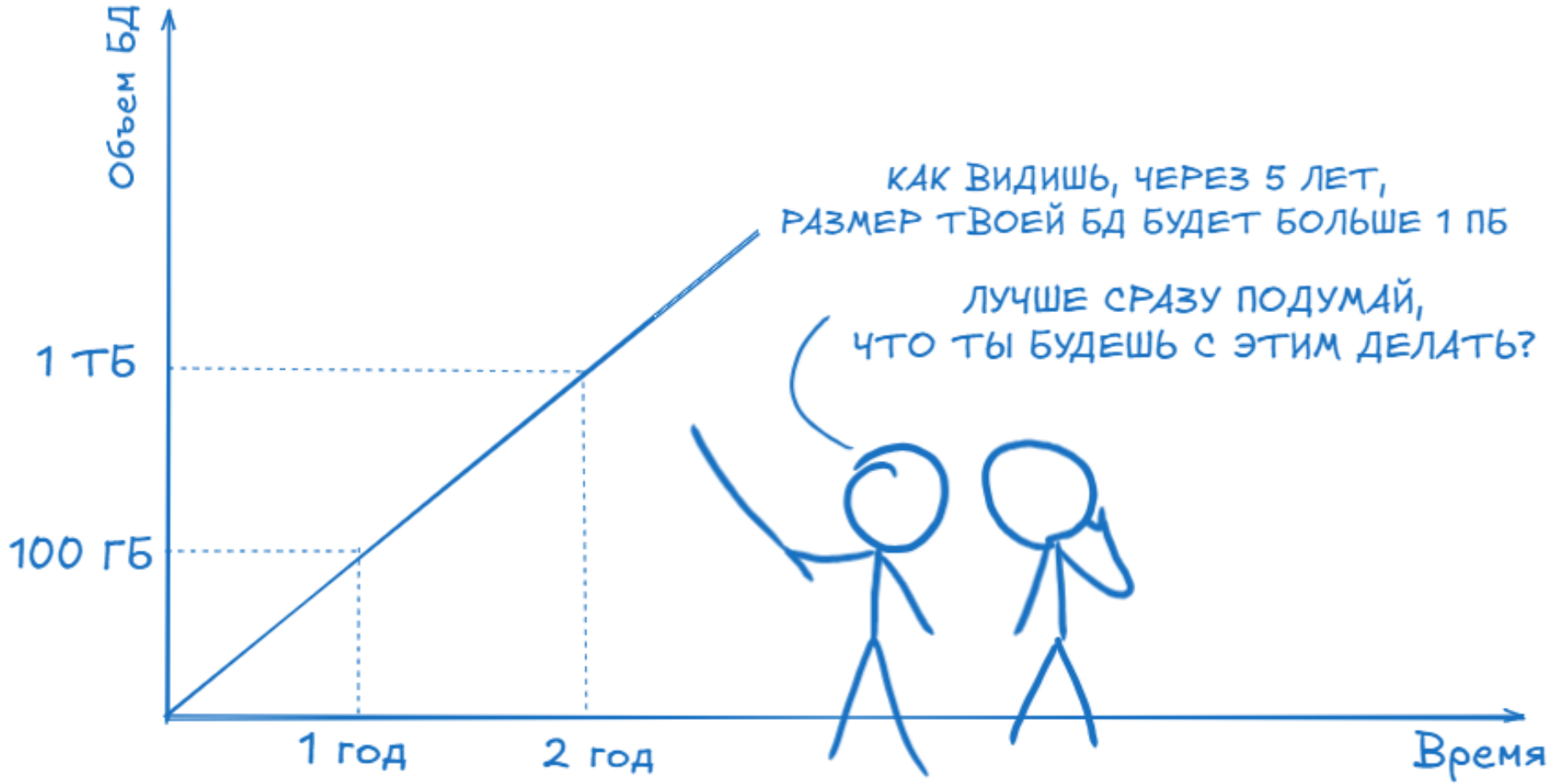
PostgresPro

Инструменты оптимизации хранения данных в PostgreSQL

Сергей Зимин
Старший технический консультант

Моё хобби — экстраполировать

PostgresPro

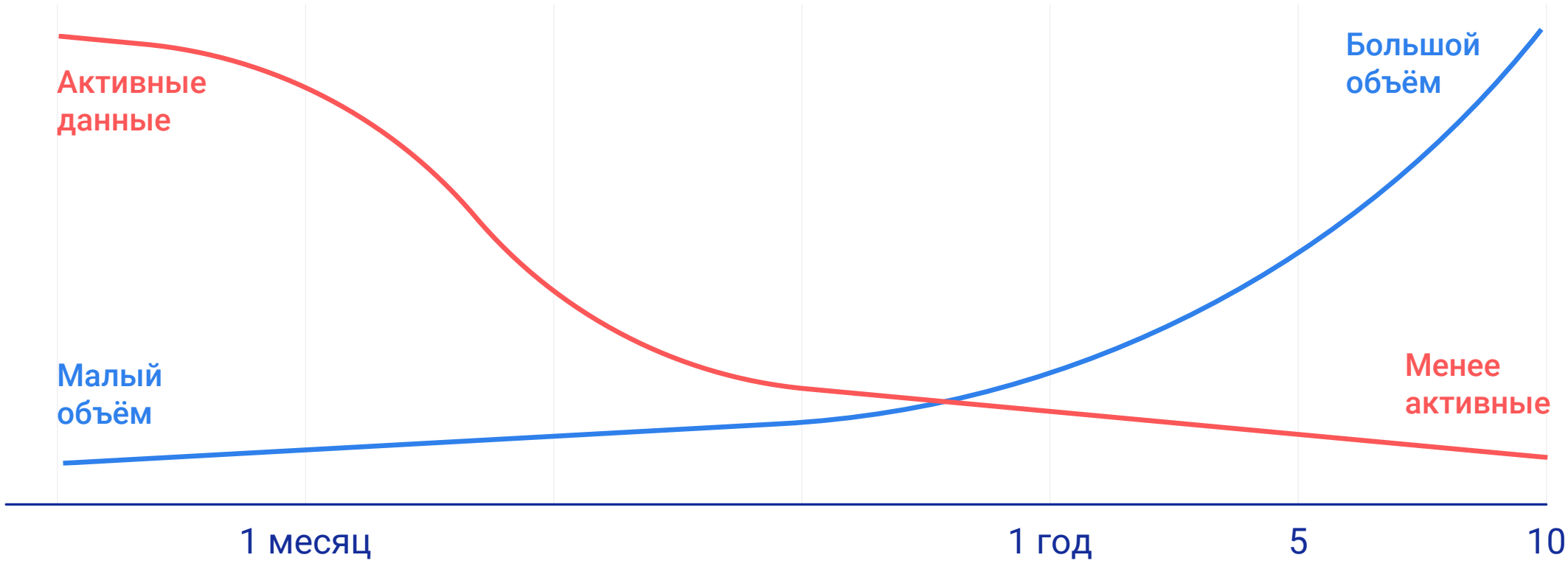


В каждой шутке лишь доля шутки...

PostgresPro

Динамика активности и объёма данных на протяжении времени

Актуальность данных и активность работы с ними со временем снижается

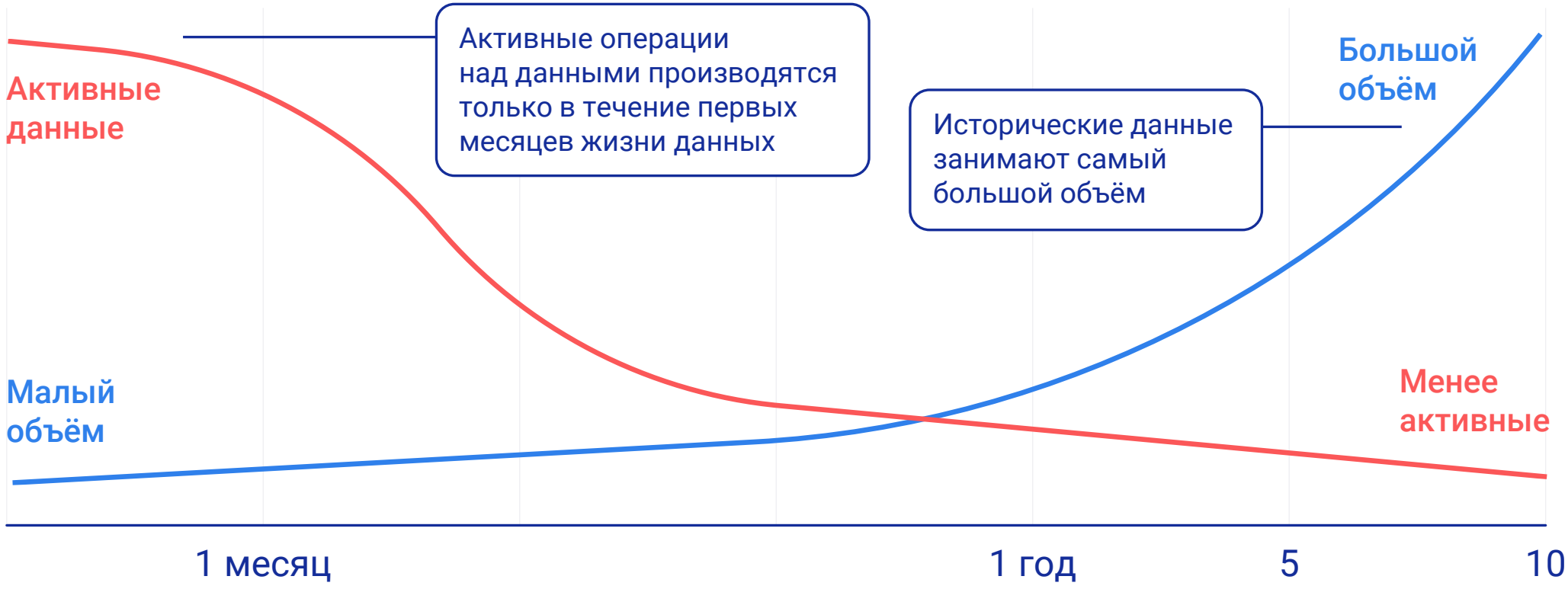


В каждой шутке лишь доля шутки...

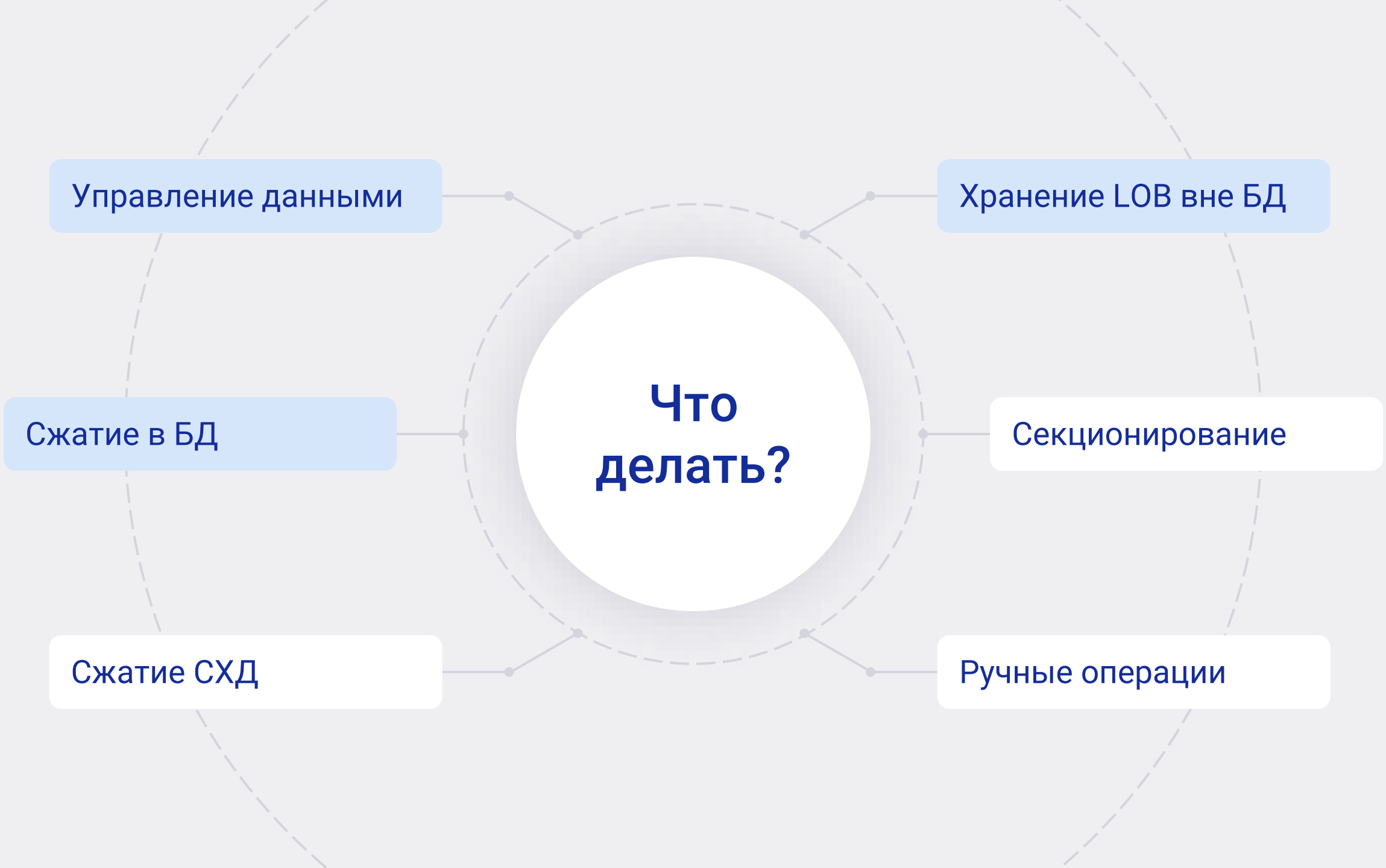
PostgresPro

Динамика активности и объёма данных на протяжении времени

Актуальность данных и активность работы с ними со временем снижается




ПРИНЦИП КВАНТОВОЙ НЕОПРЕДЕЛЕННОСТИ
ЧЕРНЫШЕВСКОГО-ГЕРЦЕНА
ОДНОВРЕМЕННО МОЖНО ТОЧНО ЗНАТЬ ЛИШЬ ОДНО:
ЛИБО
КТО ВИНОВАТ.
ЛИБО
ЧТО ДЕЛАТЬ.
/NIK-ARAGUA


Управление данными
Хранение LOB вне БД
Сжатие в БД
Что делать?
Секционирование
Сжатие СХД
Ручные операции

Сжатие в PostgreSQL

01 Сжатие TOAST, WAL и PK хорошо, но не решает все задачи

02 Сделали табличные пространства (не файловая система) с возможностью сжатия

03 Прозрачно для приложения сжимаются таблицы и индексы в табличном пространстве

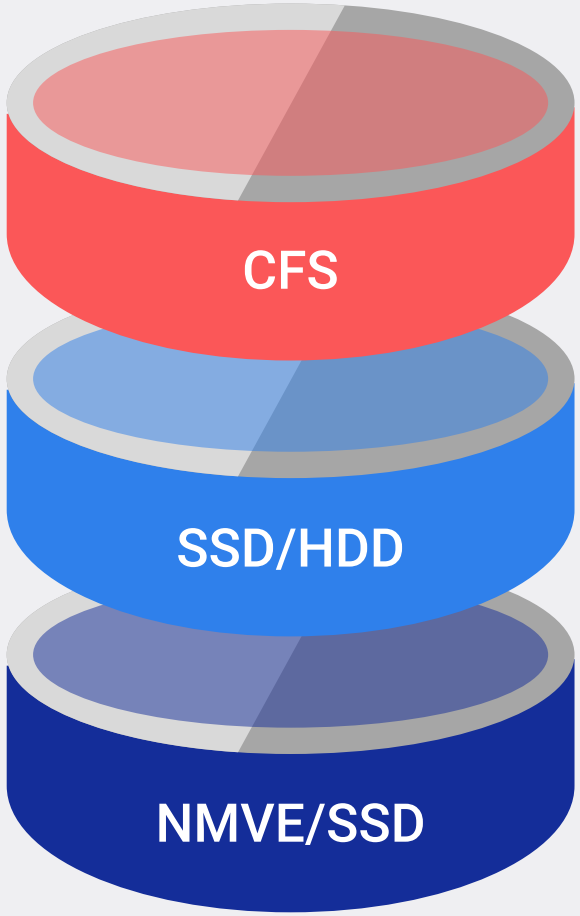
CFS — Compressed File System

PostgresPro

	CFS	WAL	TOAST	PK
Уровень сжатия	0-19	—	—	basebackup (да) probackup (0 – 22)
Поддерживаемые алгоритмы	zstd, pglz, zlib и lz4	pglz, lz4 и zstd	pglz и lz4	basebackup (gzip, lz4, zstd) probackup (zlib, lz4, zstd)
Что сжимаем	Таблицы/индексы в TBS, кроме системного каталога	WAL	TOAST, атрибуты ≥2Kb	PK
Управление	Синтаксис SQL и конфигурационные параметры	wal_compression	default_toast_compression или параметр COMPRESSION в команде CREATE/ALTER TABLE	basebackup (—compress:level) Probackup (—compress-algorithm —compress-level)
Особенности	см. ниже	Должен быть включён full_page_writes	4 стратегии: не для всех применяется сжатие Максимальный размер TOAST-таблицы – 32 TB Расходует oid	—

Information Lifecycle Management

PostgresPro



Historical

Данные не меняются (OLAP)

Редко выполняются сканирующие чтения
по колонкам

Less Active

Редко меняющиеся данные (OLTP, OLAP)

Чтения преимущественно сканирующие
по колонкам

Active

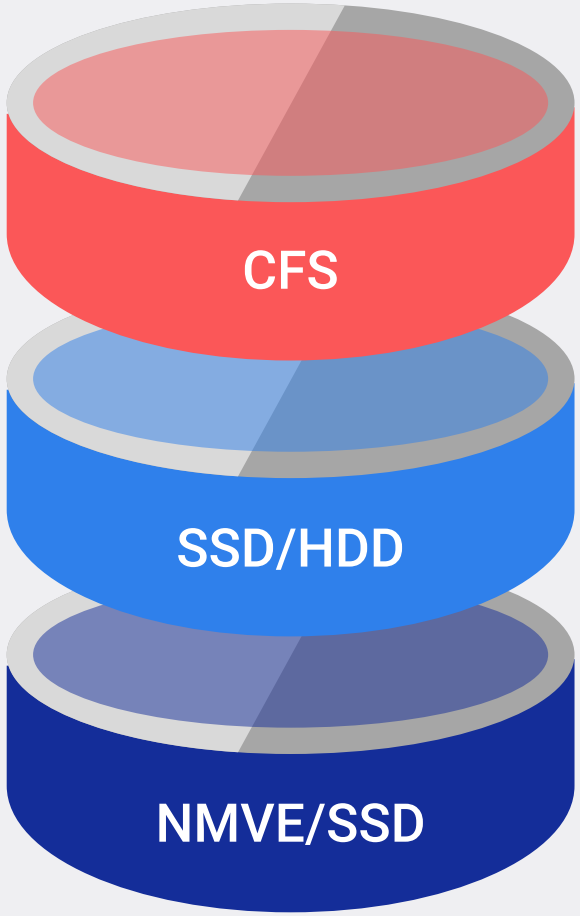
Часто меняющиеся данные (OLTP)

Очень много чтений в случайных местах

Можно реализовать самостоятельно,
НО требует рутинных, ручных операций

Information Lifecycle Management

PostgresPro



Добавили автоматизацию и статистику доступа к данным, чтобы настроил, и «оно само»

PostgresPro

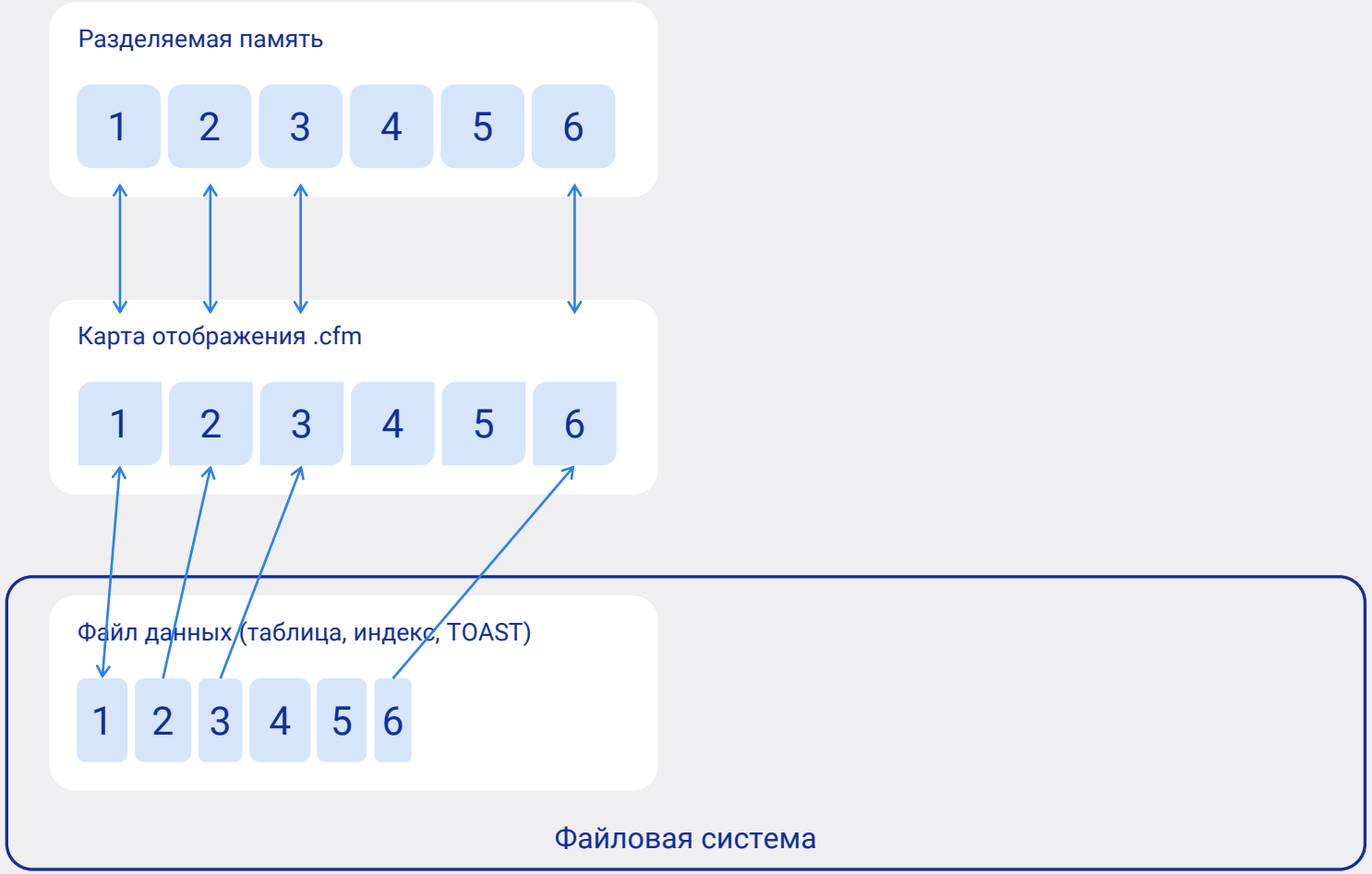
Принцип работы CFS

Без сжатия

PostgresPro

Реализация CFS

PostgresPro



Страницы в памяти 8 кб,
а на диске сжатые

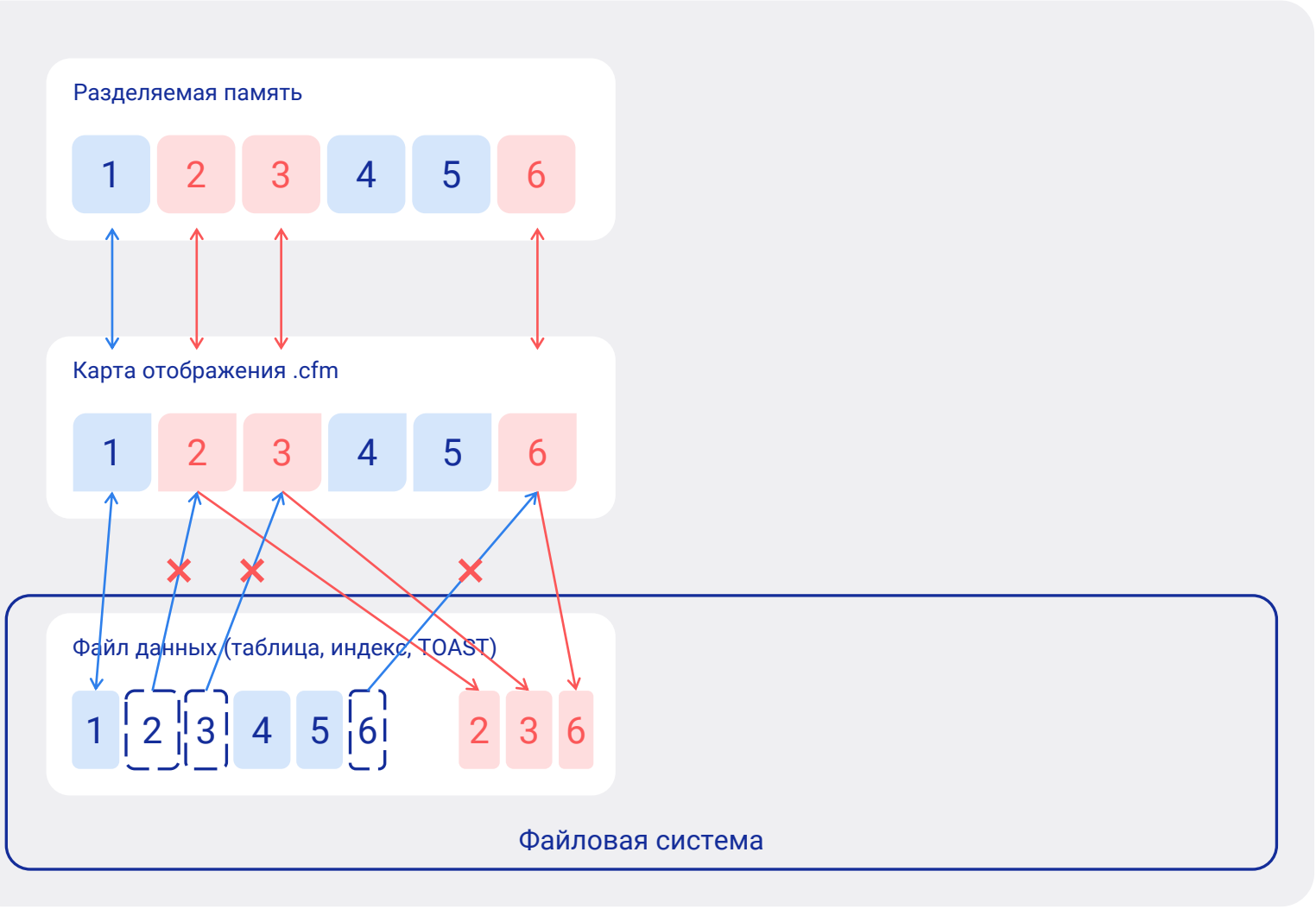
Отдельная карта отображения
для каждого файла данных

Дополнительный уровень адресации
логического адреса страницы и её
физического расположения на диске

Карта отображения хранится в памяти
и на диске

Реализация CFS: изменение данных

PostgresPro



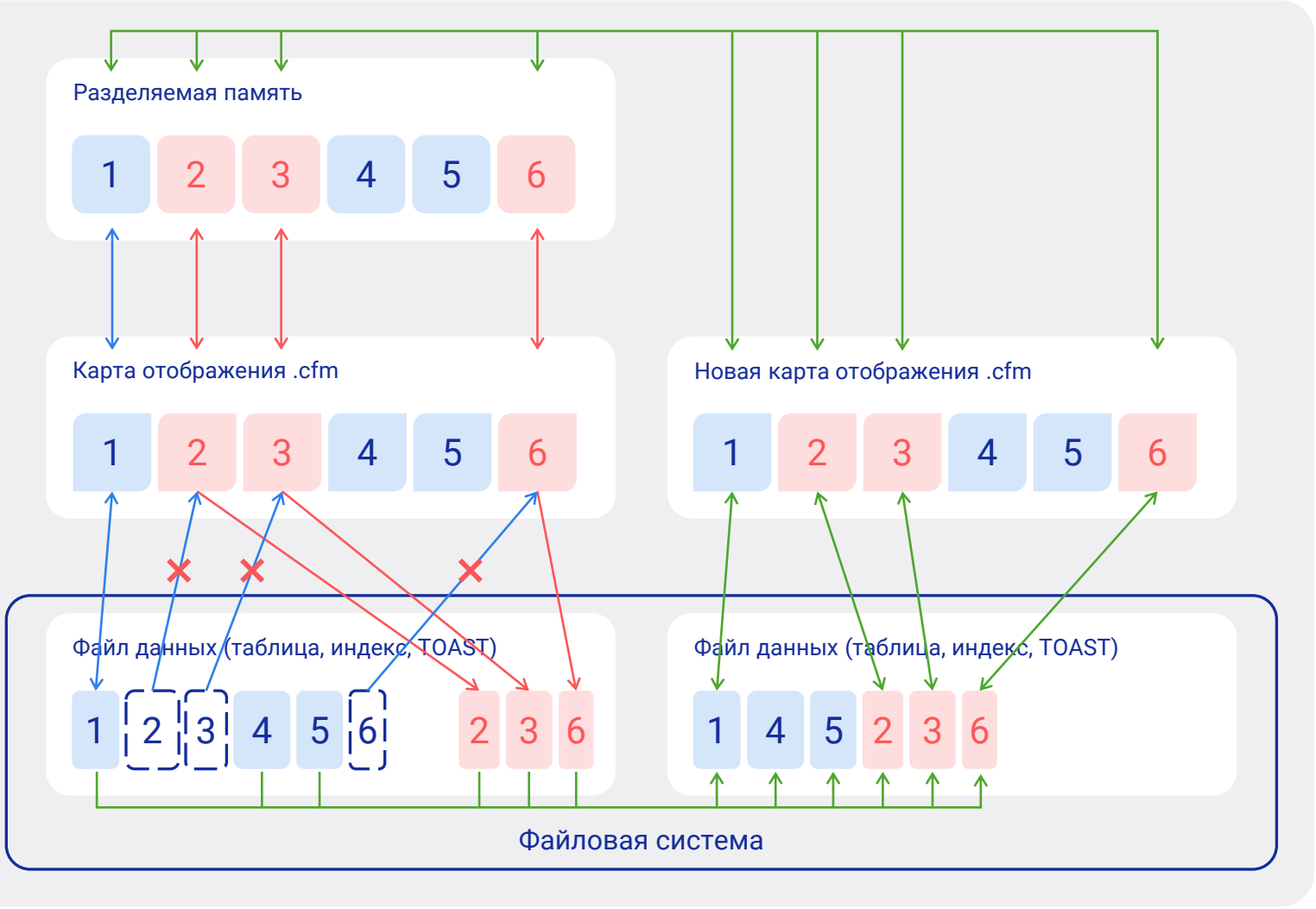
Запись изменившихся страниц в конец файла

В карту отображения вносятся соответствующие изменения

«Пустые» блоки в файле данных остаются на месте и повторно не используются до сборки мусора

Реализация CFS: сборка мусора

PostgresPro



Из-за появления «пустых» мест периодически нужна сборка мусора

Сборщик мусора CFS обрабатывает каждый файл отдельно

Сборщик мусора создаёт копии исходного файла с данными и файла отображения

Когда данные полностью сохраняются на диске, новая версия файла данных переименовывается в исходное имя

Новая карта копируется в файл, отображённый в память, и предыдущий файл отображения удаляется.

Управление сборкой мусора

PostgresPro

Сборка мусора:

- Кол-во сборщиков: `cfs_gc_workers = 1`
- Порог срабатывания сборщика: `cfs_gc_threshold = 30%`,
  до 17.4 значение по умолчанию 50%, рекомендуется его изменить

Оценка фрагментации	`cfs_fragmentation('имя таблицы')`
Запуск фрагментации	`cfs_gc_relation('имя таблицы')`
Процесс обработки сборщика мусора	`cfs_gc_activity_processed_[bytes|pages|files]()`

CFS и CHECKPOINT

PostgresPro

До

...

CHECKPOINT

После

...

До CHECKPOINT место на диске под блоки выделяется, но данных в них еще нет. То же относится к вытеснению данных из кэша буферов

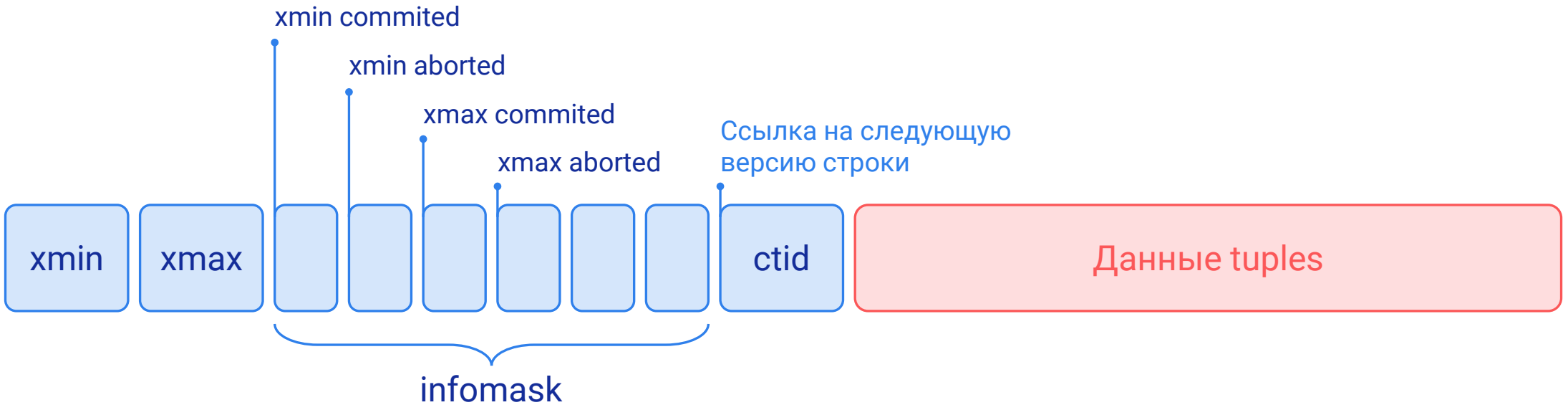
`cfs_compression_ratio()` смотрим только после CHECKPOINT

Объём на диске может «дышать», но это нормальное поведение

MVCC и побочные эффекты

PostgresPro

- Исходная транзакция не устанавливает биты, так как не знает своего финального состояния
- Если биты не установлены, то видимость кортежа без обращения к CLOG не известна
- Биты проставляются один раз, так как проверка по CLOG накладная
- Любая транзакция (даже SELECT) может менять данные в буферном кеше и породить запись в WAL и на диск



CFS и Hint Bits

PostgresPro

Ситуация: после массовых изменений
или восстановления данных

- Первый SELECT проставляет Hint Bit-ы, и все блоки будут перезаписаны в конец таблицы.
- Размер отношений и степень сжатия может «сильно» плавать в моменте

НО после сборки мусора всё станет хорошо!

Возможно, потребуется изменить временно
или в сессии `cfs_gc_threshold`
на значение меньше 30% и вызвать
`cfs_gc_relation()` вручную

CFS и bloat

PostgresPro

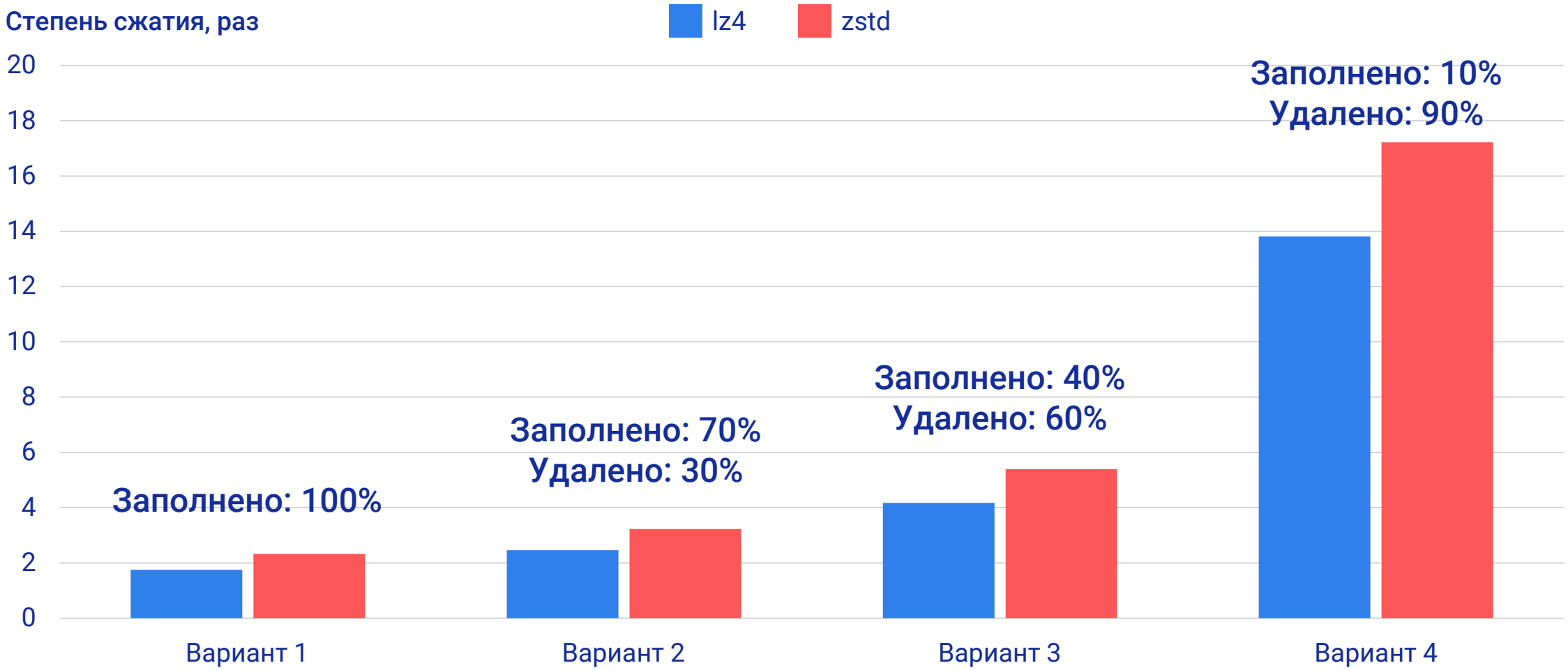
- На таблице прошли update, delete образовались dead tuples
- Vacuum или autovacuum вычистил их, в Postgres Pro Enterprise пустое место заполняется нулями
- Нули — отлично сжимаются

- CFS дает высокую степень сжатия и облегчает работу с раздутыми таблицами
- НО, получается разная степень bloat для сжатой таблицы на диске и для блоков этой же таблицы в памяти

Проверьте скрипты для вычисления bloat, они могут давать ошибочные результаты

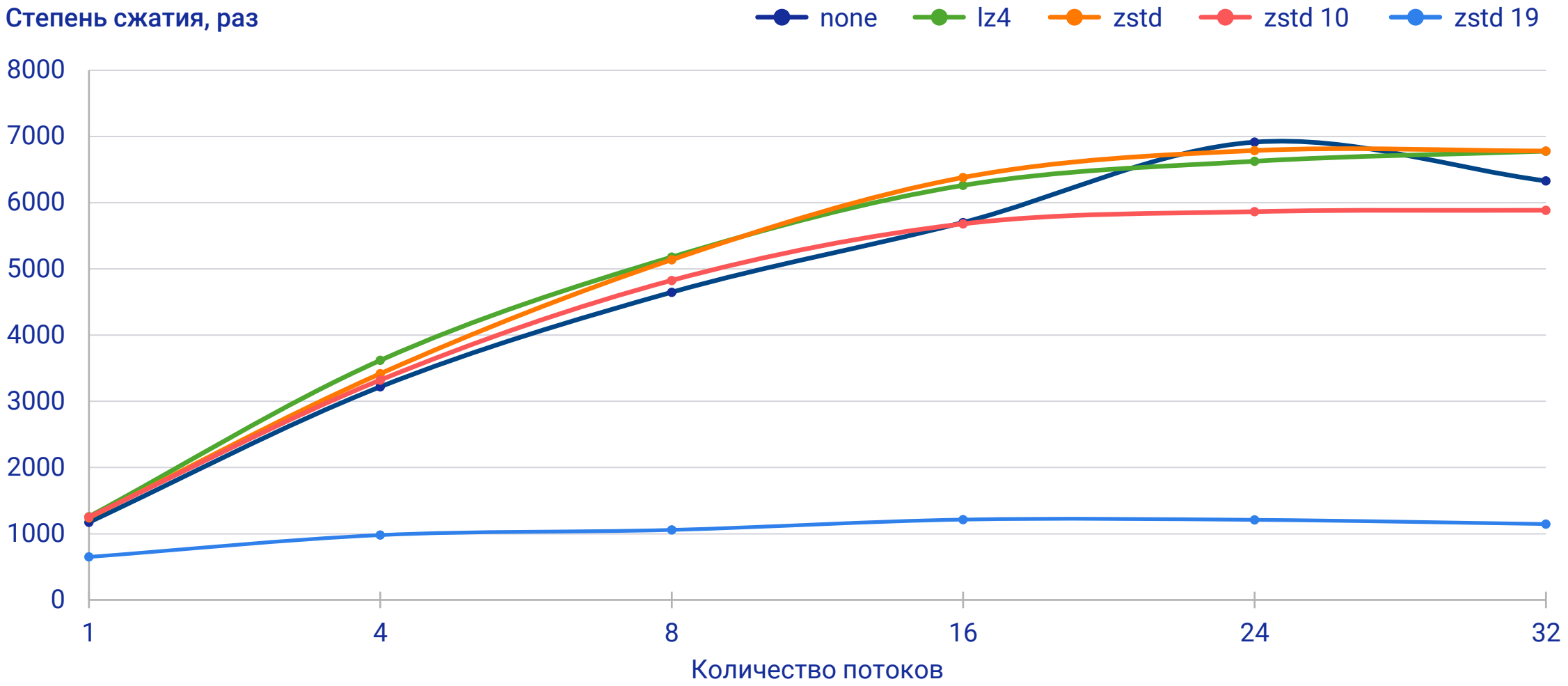
CFS и bloat

PostgresPro



Сжатие != деградации производительности

PostgresPro



Развитие и планы

Версия 17.6

Улучшили информативность вывода команды cfs_estimate:

Версия 18.1

Выйдет отложенное сжатие:

- Новый GUC: cfs_comression = 'off' – позволяет временно отключить сжатие, при массовых вставках или изменениях данных в сжатых отношениях;
- Если после вставки/изменения, нужно быстро сжать данные, то можно временно изменить cfs_gc_threshold = 0, в этом случае уже заполненные сегменты сожмутся тоже.

```
demo=# select cfs_estimate('bookings.tickets');
NOTICE:  Compression ratios and timings:
  pglz  : ratio=1.99, time=12.528 block/ms
  zlib  : ratio=2.57, time=15.340 block/ms
  lz4   : ratio=1.97, time=108.669 block/ms
  zstd  : ratio=2.62, time=38.612 block/ms

Best ratio: "zstd" with 2.62
Best speed: "lz4" with 108.67 blocks/ms


    cfs_estimate
--------------------
 2.6184461451715433
(1 row)
```

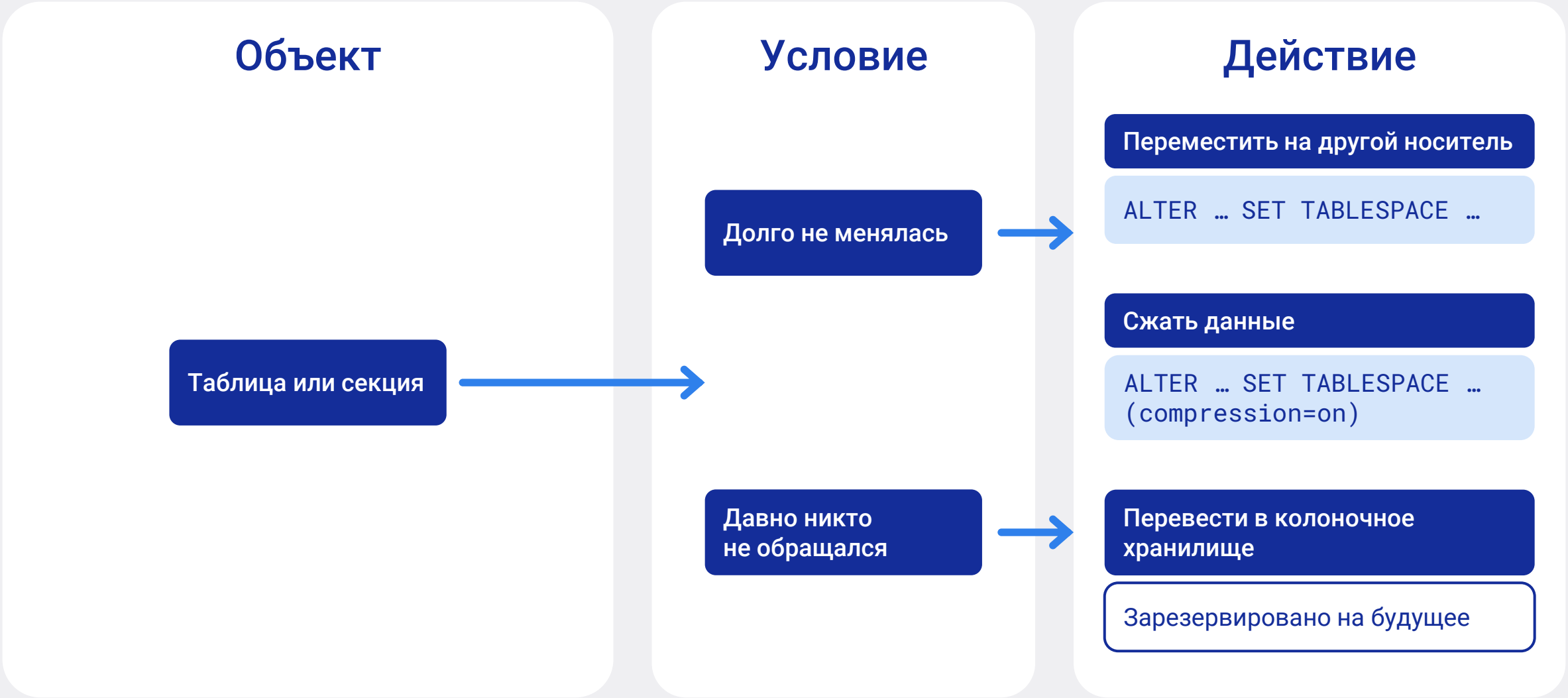
Версия 17.6

PostgresPro

Принцип работы ILM

ILM автоматизация

PostgresPro



ILM автоматизация

PostgresPro

Индексы перемещаются вместе с соответствующими таблицами

Объект			Политика		Действие	
nspname	relname	relkind	rule_type	period	action	parameter
app_schema	sales_table_section_q1_2021		NO_MODIFICATION	3 months	ALTER_TS	low_cost_sales_tablespace
app_schema	sales_table_section_q1_2021		NO_ACCESS	12 months	ALTER_TS	cfs_archive_sales_tablespace
app_schema	sales_table_section_q1_2021	i	NO_ACCESS	12 months	ALTER_TS	cfs_archive_sales_tablespace
app_schema	sales_table_section_q1_2021		NO_MODIFICATION	18 months	ALTER_TS	cfs_archive_sales_tablespace
app_schema	sales_table_section_q1_2021	i	NO_MODIFICATION	18 months	ALTER_TS	cfs_archive_sales_tablespace
...	...	...	...	...	...	...

ILM автоматизация

PostgresPro

Объект			Условие		Действие	
nspname	relname	relkind	rule_type	period	action	parameter
app_schema	sales_table_section_q1_2021		NO_MODIFICATION	3 months	ALTER_TS	low_cost_sales_tablespace
app_schema	sales_table_section_q1_2021		NO_ACCESS	12 months	ALTER_TS	cfs_archive_sales_tablespace
app_schema	sales_table_section_q1_2021			2 months	ALTER_TS	cfs_archive_sales_tablespace
app_schema	sales_table_section_q1_2021			3 months	ALTER_TS	cfs_archive_sales_tablespace
app_schema	sales_table_section_q1_2021	i	NO_MODIFICATION	18 months	ALTER_TS	cfs_archive_sales_tablespace
...	...	...	...	...	...	...

Правила проверяются в порядке убывания period

ILM автоматизация

PostgresPro

Объект			Условие		Действие	
nspname	relname	relkind	rule_type	period	action	parameter
app_schema	sales_table_section_q1_2021		NO_MODIFICATION	3 months	ALTER_TS	low_cost_sales_tablespace
app_schema	sales_table_section_q1_2021		NO_ACCESS	12 months	ALTER_TS	cfs_archive_sales_tablespace
app_schema	sales_table_section_q1_2021	i		nths	ALTER_TS	cfs_archive_sales_tablespace
app_schema	sales_table_section_q1_2021			nths	ALTER_TS	cfs_archive_sales_tablespace
app_schema	sales_table_section_q1_2021	i	NO_MODIFICATION	18 months	ALTER_TS	cfs_archive_sales_tablespace
...	...	...	...	...	...	...

COLUMNAR зарезервировано для реализации в будущем

ILM автоматизация

Less Active Historical

PostgresPro

Объект

Условие

Действие

nspname	relname	relkind	rule_type	period	action	parameter
app_schema	sales_table_section_q1_2021		NO_MODIFICATION	3 months	ALTER_TS	low_cost_sales_tablespace
app_schema	sales_table_section_q1_2021		NO_ACCESS	12 months	ALTER_TS	cfs_archive_sales_tablespace
app_schema	sales_table_section_q1_2021	i	NO_ACCESS	12 months	ALTER_TS	cfs_archive_sales_tablespace
app_schema	sales_table_section_q1_2021		NO_MODIFICATION	18 months	ALTER_TS	cfs_archive_sales_tablespace
app_schema	sales_table_section_q1_2021	i	NO_MODIFICATION	18 months	ALTER_TS	cfs_archive_sales_tablespace
...	...	...	...	...	...	...

Обработка условий

PostgresPro

Добавили статистику обращений к таблице в расширение pgpro_usage

В рамках задачи «Поиска неиспользуемых привилегий»

Собираем 2 вида статистики:

- Долгосрочная для pgpro_ilm;
- Статистика для pgpro_usage

Правила заданы и хранятся в табличной форме в памяти и на диске

Статистика
для Information
Lifecycle Management

Статистика
для поиска
неиспользуемых
привилегий

Расширенная статистика (pgpro_usage)

```
# select * from pg_stat_all_tables_last_usage \gx
- [ RECORD 1 ] --+-------------------------
userid           | 7319
username         | user1
nspname          | app_schema
relid            | 16548
relname          | sales_table_section_q1_2021
last_read        | 2022-08-17 16:47:18.478406+03
last_insert      | 2021-03-31 23:59:49.134439+03
last_update      | 2021-03-31 23:48:13.409240+03
last_delete      | 2021-03-30 18:25:34.094744+03
last_truncate    |
```

NO_ACCESS

NO_MODIFICATION

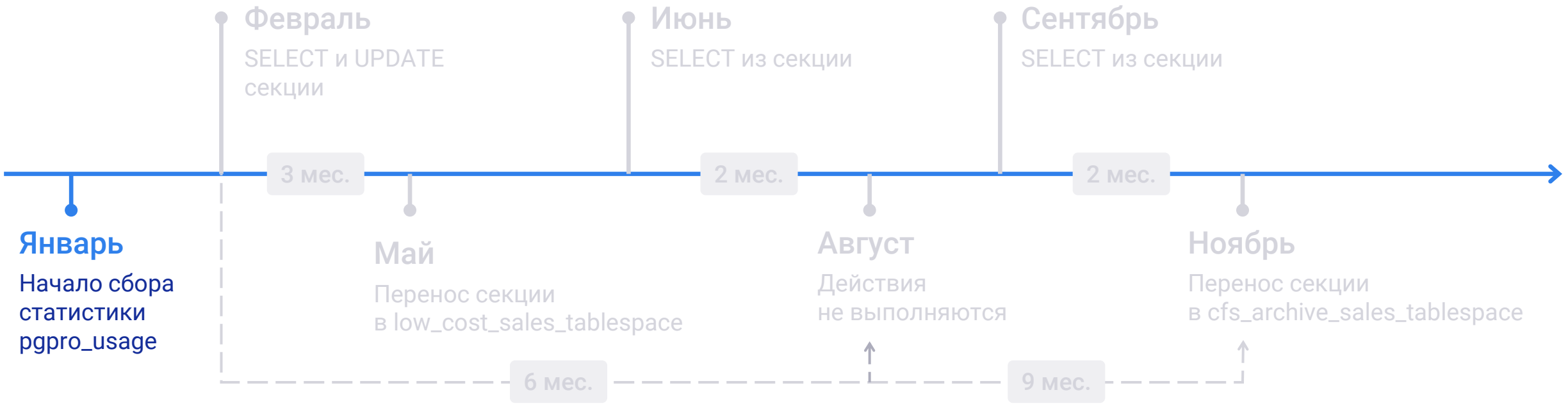
pgpro_usage_reset(); — по умолчанию запускается со значением false, при запуске со значением true сбросится вся статистика

Пример обработки правил ILM

В конце каждого месяца
запускается обработка правил

PostgresPro

nspname	relname	rule_type	period	action	parameter	
app_schema	sales_table_section_q1_2021	NO_MODIFICATION	3 months	ALTER_TS	low_cost_sales_tablespace	Less Active
app_schema	sales_table_section_q1_2021	NO_ACCESS	6 months	ALTER_TS	cfs_archive_sales_tablespace	Historical
app_schema	sales_table_section_q1_2021	NO_MODIFICATION	9 months	ALTER_TS	cfs_archive_sales_tablespace	Historical

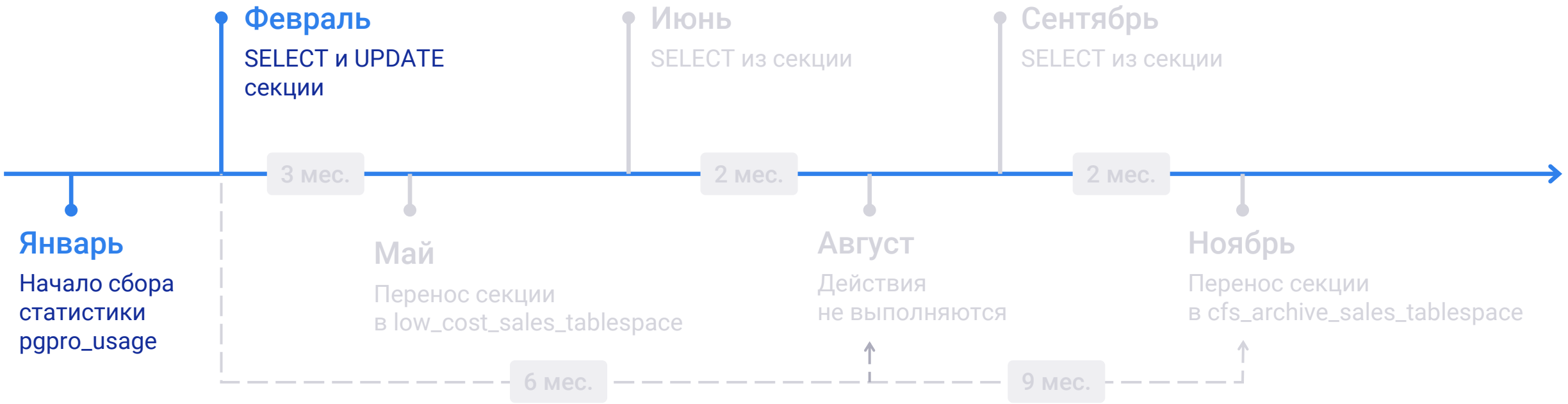


Пример обработки правил ILM

В конце каждого месяца
запускается обработка правил

PostgresPro

nspname	relname	rule_type	period	action	parameter	
app_schema	sales_table_section_q1_2021	NO_MODIFICATION	3 months	ALTER_TS	low_cost_sales_tablespace	Less Active
app_schema	sales_table_section_q1_2021	NO_ACCESS	6 months	ALTER_TS	cfs_archive_sales_tablespace	Historical
app_schema	sales_table_section_q1_2021	NO_MODIFICATION	9 months	ALTER_TS	cfs_archive_sales_tablespace	Historical

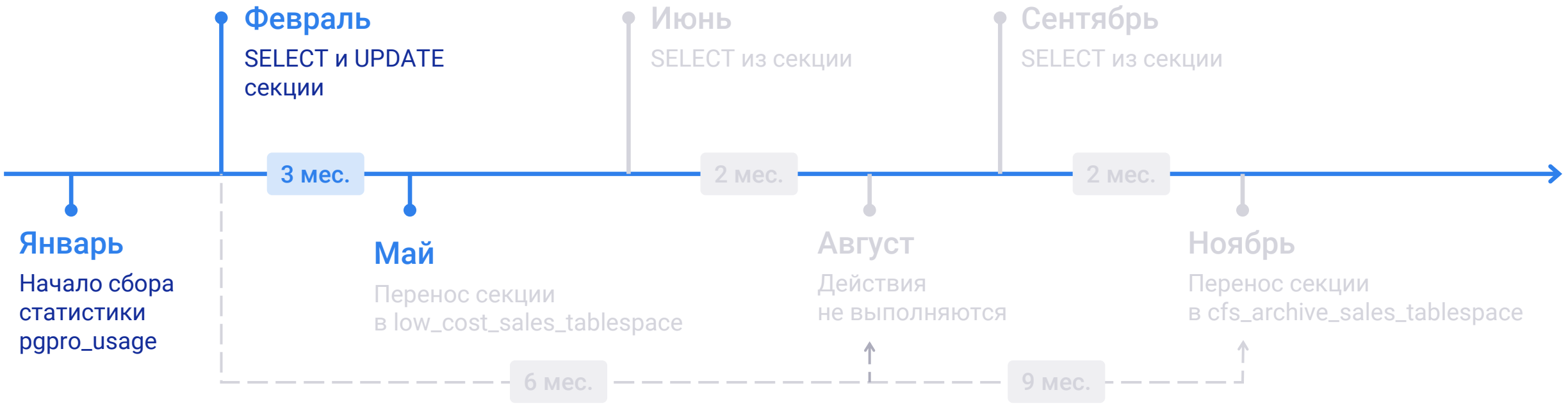


Пример обработки правил ILM

В конце каждого месяца
запускается обработка правил

PostgresPro

nspname	relname	rule_type	period	action	parameter	
app_schema	sales_table_section_q1_2021	NO_MODIFICATION	3 months	ALTER_TS	low_cost_sales_tablespace	Less Active
app_schema	sales_table_section_q1_2021	NO_ACCESS	6 months	ALTER_TS	cfs_archive_sales_tablespace	Historical
app_schema	sales_table_section_q1_2021	NO_MODIFICATION	9 months	ALTER_TS	cfs_archive_sales_tablespace	Historical

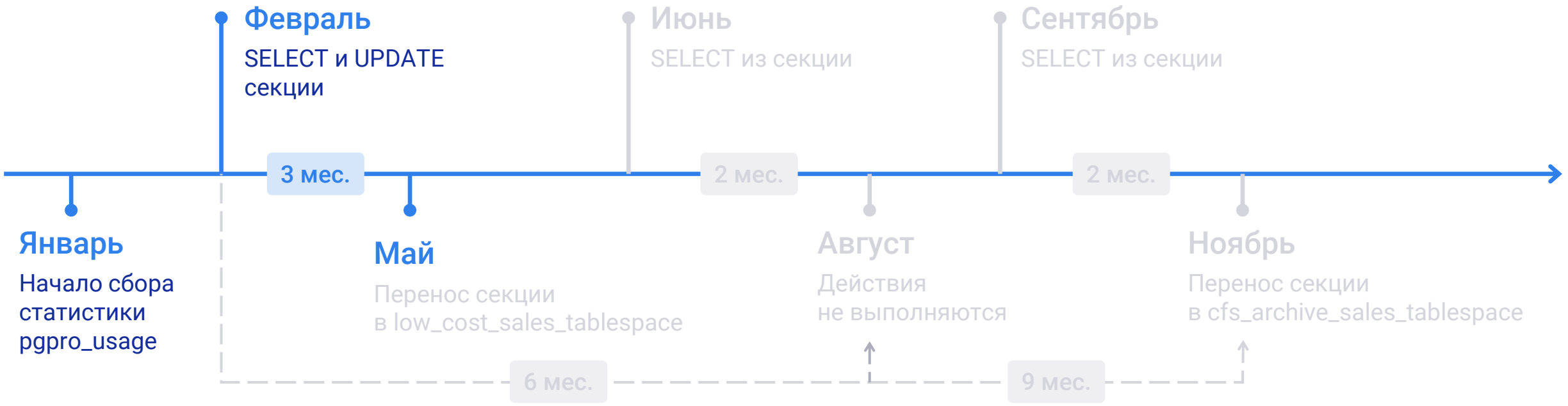


Пример обработки правил ILM

В конце каждого месяца
запускается обработка правил

PostgresPro

nspname	relname	rule_type	period	action	parameter	
app_schema	sales_table_section_q1_2021	NO_MODIFICATION	3 months	ALTER_TS	low_cost_sales_tablespace	Less Active
app_schema	sales_table_section_q1_2021	NO_ACCESS	6 months	ALTER_TS	cfs_archive_sales_tablespace	Historical
app_schema	sales_table_section_q1_2021	NO_MODIFICATION	9 months	ALTER_TS	cfs_archive_sales_tablespace	Historical

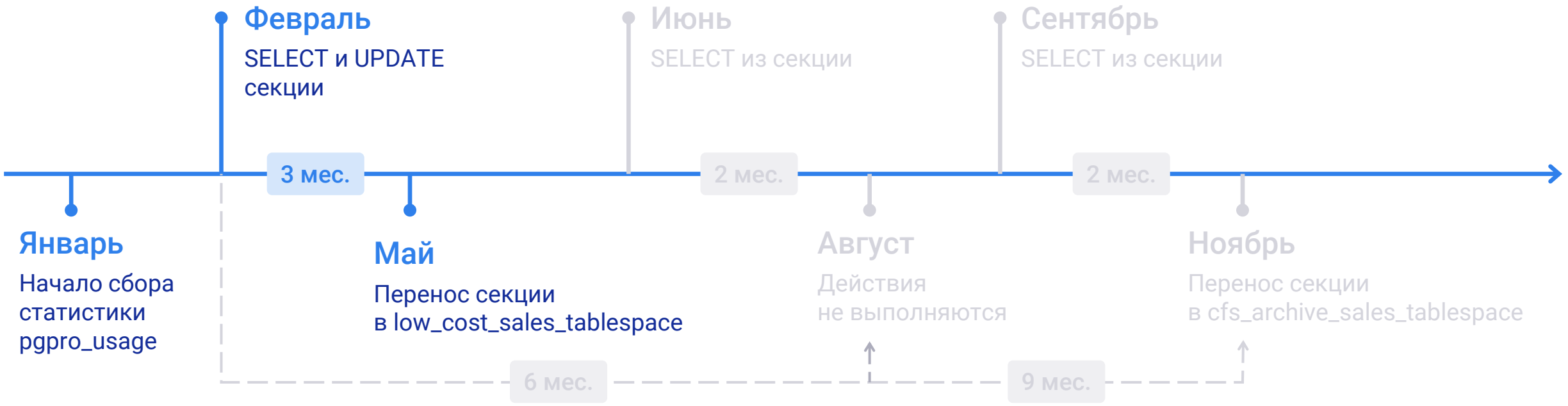


Пример обработки правил ILM

В конце каждого месяца
запускается обработка правил

PostgresPro

nspname	relname	rule_type	period	action	parameter	
app_schema	sales_table_section_q1_2021	NO_MODIFICATION	3 months	ALTER_TS	low_cost_sales_tablespace	Less Active
app_schema	sales_table_section_q1_2021	NO_ACCESS	6 months	ALTER_TS	cfs_archive_sales_tablespace	Historical
app_schema	sales_table_section_q1_2021	NO_MODIFICATION	9 months	ALTER_TS	cfs_archive_sales_tablespace	Historical

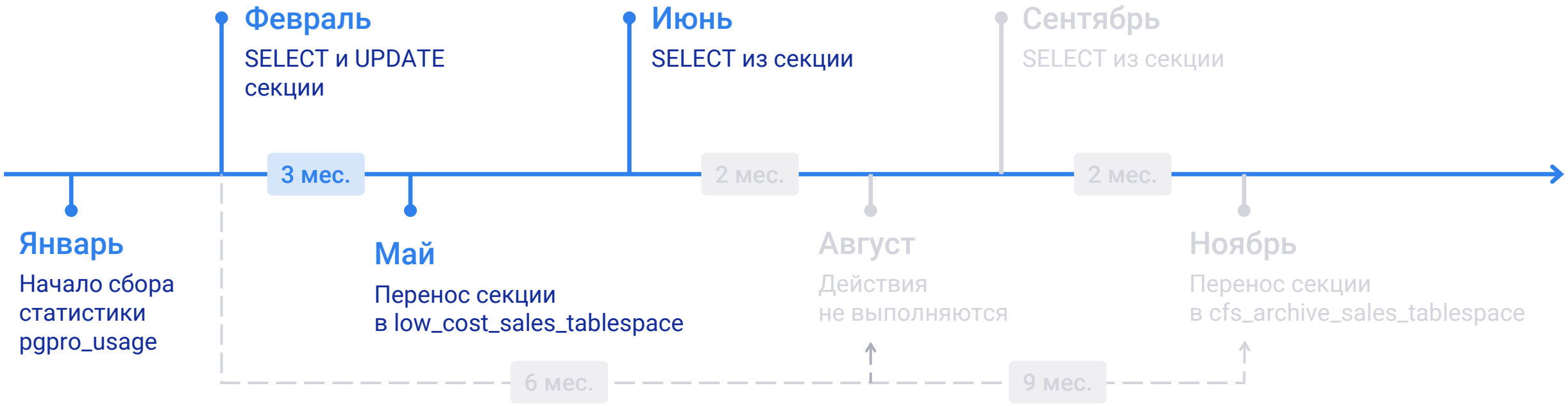


Пример обработки правил ILM

В конце каждого месяца
запускается обработка правил

PostgresPro

nspname	relname	rule_type	period	action	parameter	
app_schema	sales_table_section_q1_2021	NO_MODIFICATION	3 months	ALTER_TS	low_cost_sales_tablespace	Less Active
app_schema	sales_table_section_q1_2021	NO_ACCESS	6 months	ALTER_TS	cfs_archive_sales_tablespace	Historical
app_schema	sales_table_section_q1_2021	NO_MODIFICATION	9 months	ALTER_TS	cfs_archive_sales_tablespace	Historical

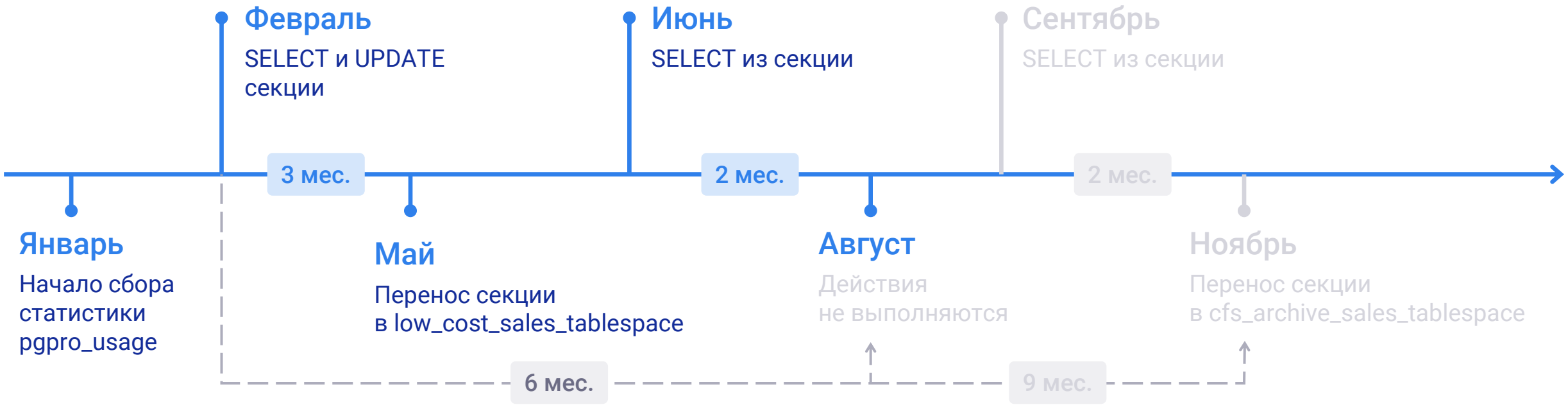


Пример обработки правил ILM

В конце каждого месяца
запускается обработка правил

PostgresPro

nspname	relname	rule_type	period	action	parameter	
app_schema	sales_table_section_q1_2021	NO_MODIFICATION	3 months	ALTER_TS	low_cost_sales_tablespace	Less Active
app_schema	sales_table_section_q1_2021	NO_ACCESS	6 months	ALTER_TS	cfs_archive_sales_tablespace	Historical
app_schema	sales_table_section_q1_2021	NO_MODIFICATION	9 months	ALTER_TS	cfs_archive_sales_tablespace	Historical

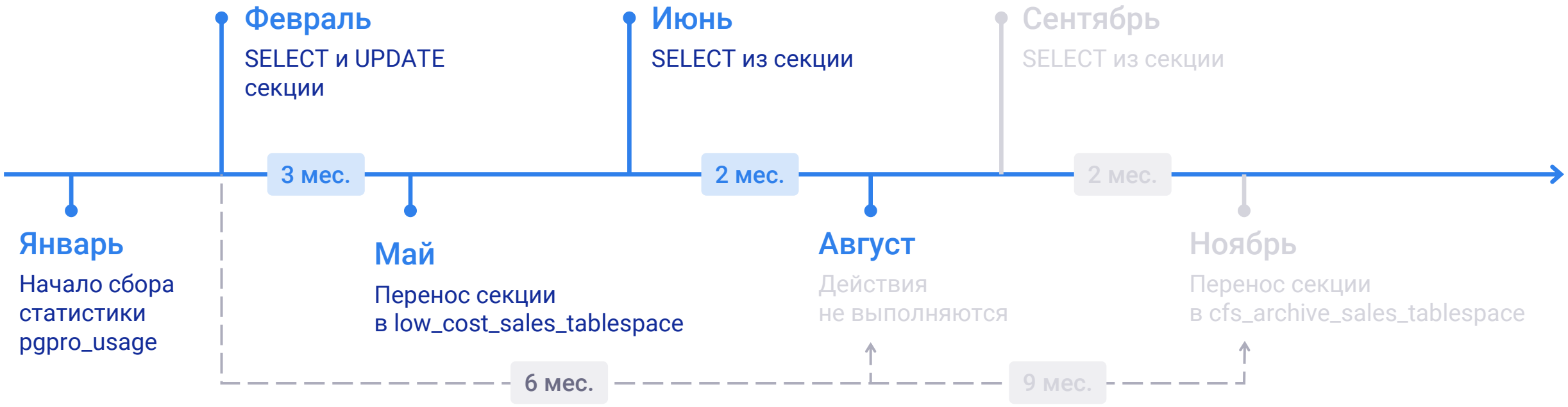


Пример обработки правил ILM

В конце каждого месяца
запускается обработка правил

PostgresPro

nspname	relname	rule_type	period	action	parameter	
app_schema	sales_table_section_q1_2021	NO_MODIFICATION	3 months	ALTER_TS	low_cost_sales_tablespace	Less Active
app_schema	sales_table_section_q1_2021	NO_ACCESS	6 months	ALTER_TS	cfs_archive_sales_tablespace	Historical
app_schema	sales_table_section_q1_2021	NO_MODIFICATION	9 months	ALTER_TS	cfs_archive_sales_tablespace	Historical

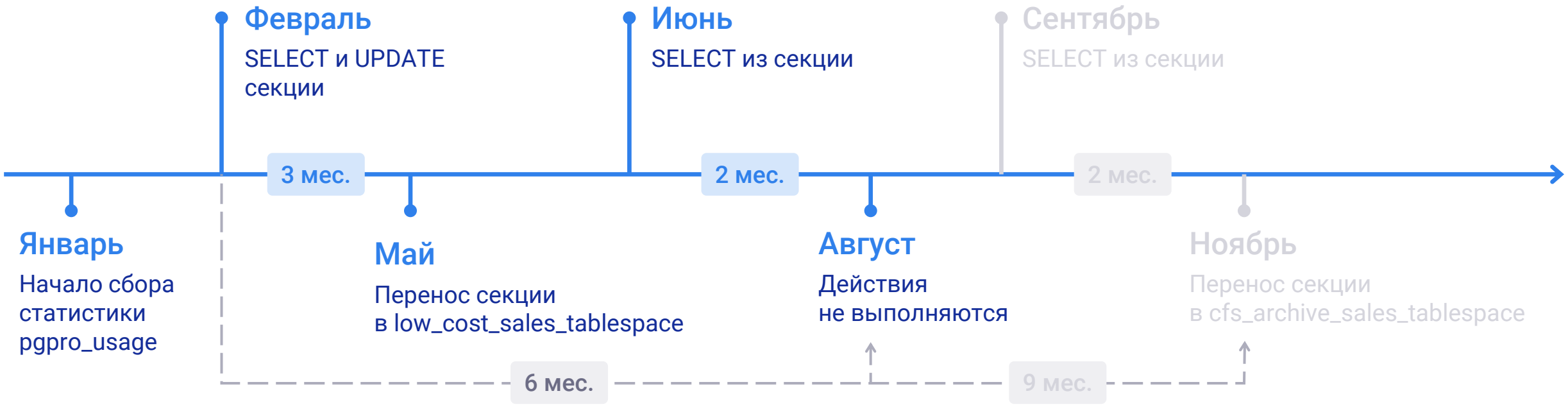


Пример обработки правил ILM

В конце каждого месяца
запускается обработка правил

PostgresPro

nspname	relname	rule_type	period	action	parameter	
app_schema	sales_table_section_q1_2021	NO_MODIFICATION	3 months	ALTER_TS	low_cost_sales_tablespace	Less Active
app_schema	sales_table_section_q1_2021	NO_ACCESS	6 months	ALTER_TS	cfs_archive_sales_tablespace	Historical
app_schema	sales_table_section_q1_2021	NO_MODIFICATION	9 months	ALTER_TS	cfs_archive_sales_tablespace	Historical

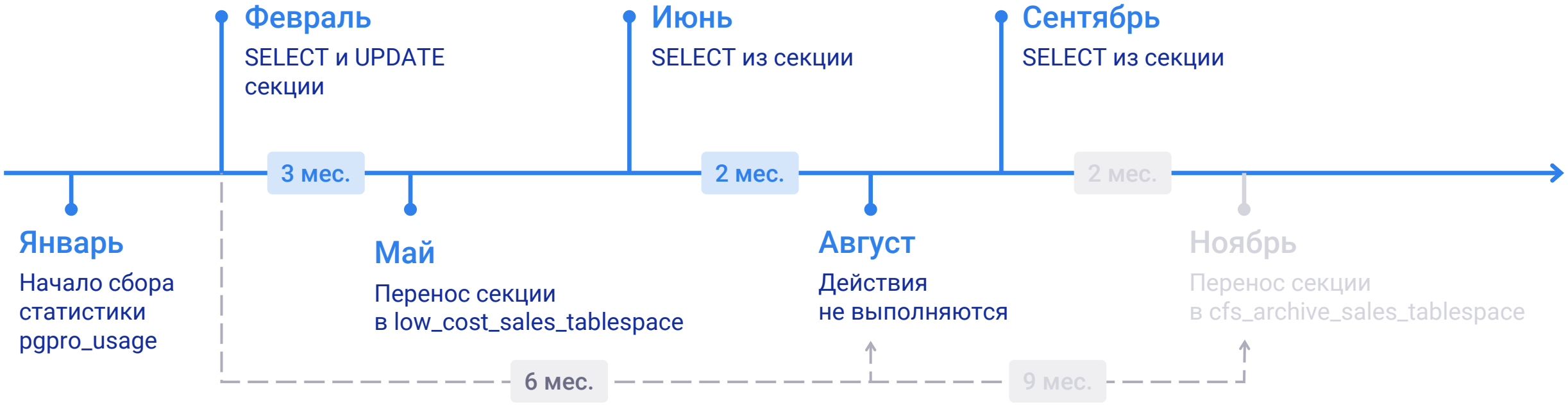


Пример обработки правил ILM

В конце каждого месяца
запускается обработка правил

PostgresPro

nspname	relname	rule_type	period	action	parameter	
app_schema	sales_table_section_q1_2021	NO_MODIFICATION	3 months	ALTER_TS	low_cost_sales_tablespace	Less Active
app_schema	sales_table_section_q1_2021	NO_ACCESS	6 months	ALTER_TS	cfs_archive_sales_tablespace	Historical
app_schema	sales_table_section_q1_2021	NO_MODIFICATION	9 months	ALTER_TS	cfs_archive_sales_tablespace	Historical

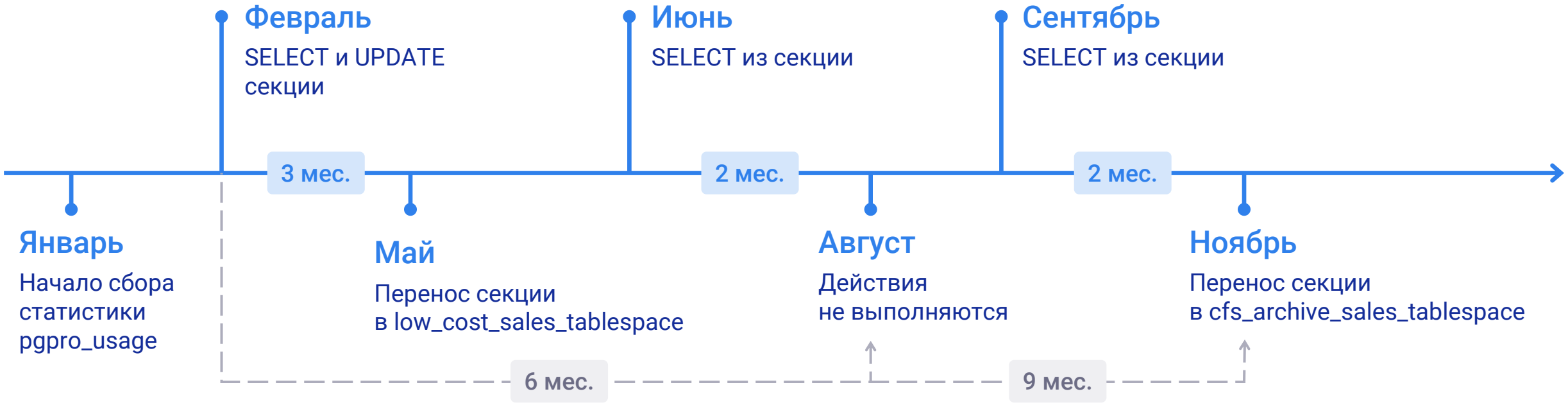


Пример обработки правил ILM

В конце каждого месяца
запускается обработка правил

PostgresPro

nspname	relname	rule_type	period	action	parameter	
app_schema	sales_table_section_q1_2021	NO_MODIFICATION	3 months	ALTER_TS	low_cost_sales_tablespace	Less Active
app_schema	sales_table_section_q1_2021	NO_ACCESS	6 months	ALTER_TS	cfs_archive_sales_tablespace	Historical
app_schema	sales_table_section_q1_2021	NO_MODIFICATION	9 months	ALTER_TS	cfs_archive_sales_tablespace	Historical



Вопросы, на которые нужно ответить

PostgresPro

ЧТО переносим и какие пользователи не должны оказывать влияние на работу ILM?

- Служебные пользователи;
- Аудиторы;
- Отчётность;
- Пакетные операции;
- и т. д.

ГДЕ будут храниться данные после переноса?

Подготовить необходимые табличные пространства

КОГДА и в каких случаях переносим данные?

Разразработать регламент управление жизненным циклом данных

На что обратить внимание

PostgresPro

- Секций секционированных таблиц наследуют правила от родительской таблицы
- Индексы переносятся вместе с таблицей/секцией, если нет отдельных правил для индексов
- Если таблица уже была перенесена, то повторно правило не обрабатывается
- Запускайте обработку правил ночью или иной период с наименьшей нагрузкой
- Если ALTER TABLE завершилась ошибкой, то повторный вызов process_rules повторит операцию снова

PostgresPro

Бонус

Расширение Bfile

Расширение Bfile уменьшает объем БД

Если в ней хранятся медиафайлы/LOB

- Поддерживает разграничение доступа к файлам;
- Можно создать отдельные каталоги для каждого пользователя;
- Поддерживает работу через собственный SQL-интерфейс или через обёртку dbms_lob (аналог Oracle DBMS_LOB) для работы с bfile;

```
# mkdir "/tmp/bfiles"
CREATE EXTENSION pgpro_bfile;
-- Создание каталога в базе данных для хранения bfile:
SELECT bfile_directory_create('BFILE_DATA',
'/tmp/bfiles');
-- Создание файла bfile.data в файловой системе и
добавление в него значения '0123456789':
SELECT bfile_write_direct(bfile_make('BFILE_DATA',
'bfile.data'), '0123456789');
-- Создание таблицы bfile и добавление в неё одной
записи, ссылающейся на этот файл:
CREATE TABLE user_doc(id int, bf bfile);
INSERT INTO bfile_table VALUES (1,
bfile_make('BFILE_DATA', 'bfile.data'));
-- Создание пользователя для  обращения к файлу:
CREATE USER bf_test_user;
-- Предоставление пользователю bf_test_user права чтения
и записи в файл BFILE_DATA:
SELECT bfile_grant_directory('BFILE_DATA',
'bf_test_user', 3);
-- Открытие файла для чтения и записи под пользователем
db_test_user:
SELECT bfile_open(bf, 3) FROM user_doc WHERE id = 1;
→61
-- Эта строка символов будет записана в конец файла:
SELECT bfile_write( 61, '_suffix');
SELECT encode( bfile_read(61) , 'escape');
→123456789_suffix
SELECT bfile_close(61);
```

PostgresPro

Отвечу на ваши вопросы!

Сергей Зимин
Старший технический консультант