



Postgres Pro — отечественная **коммерческая** СУБД

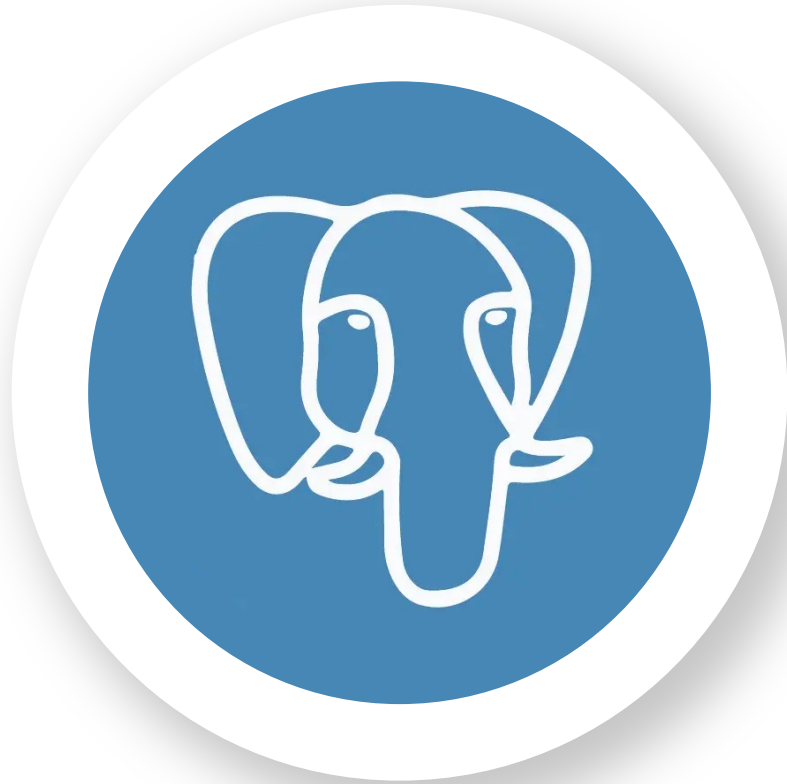
Марк Ривкин

Начальник отдела технического консалтинга,
Postgres Professional

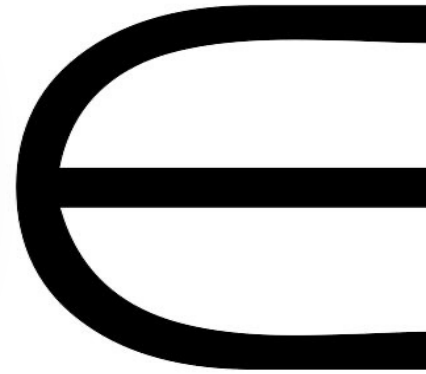
Что такое Postgres Pro Enterprise

- **Отечественная коммерческая объектно-реляционная СУБД** для высоконагруженных систем крупных предприятий
- **Богатая функциональность** (SQL, таблицы, индексы, MV, view, последовательности, функции, процедуры, триггеры, ACID, роли/привилегии, ограничения целостности, BLOB, оконные функции и т. д.) + **ПАКЕТЫ**
- **Разработчики и техническая поддержка в России**
- **Сертификат ФСТЭК Г1, уровень доверия УД 4** (защита конфиденциальной информации, у Oracle и MS — УД ниже и = 6)
- **В реестре** российских программных продуктов
- **Регулярное слияние** с ванильной СУБД PostgreSQL
- **Расширяемый код**

Postgres Pro включает все возможности PostgreSQL



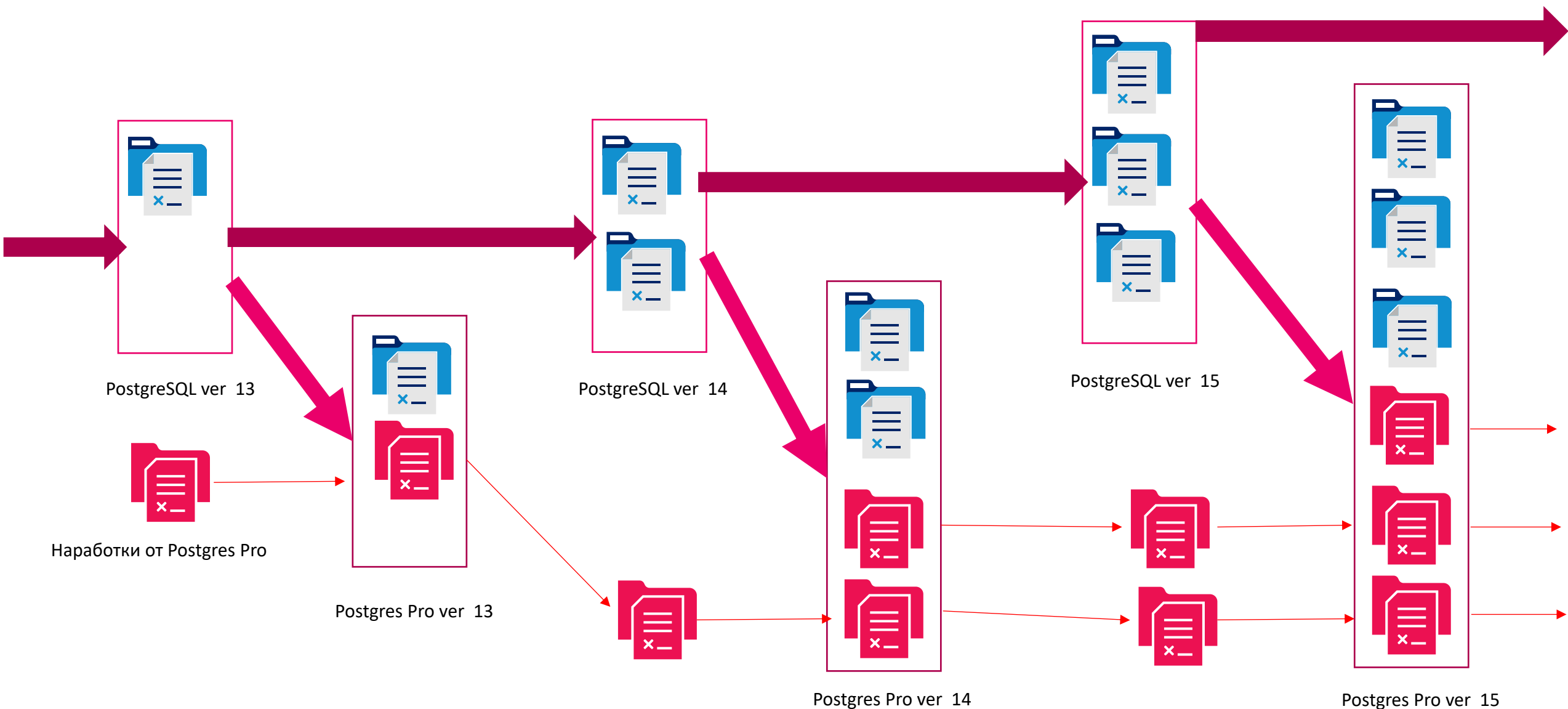
PostgreSQL



Postgres Pro

Слияние версий PostgreSQL и Postgres Pro Enterprise

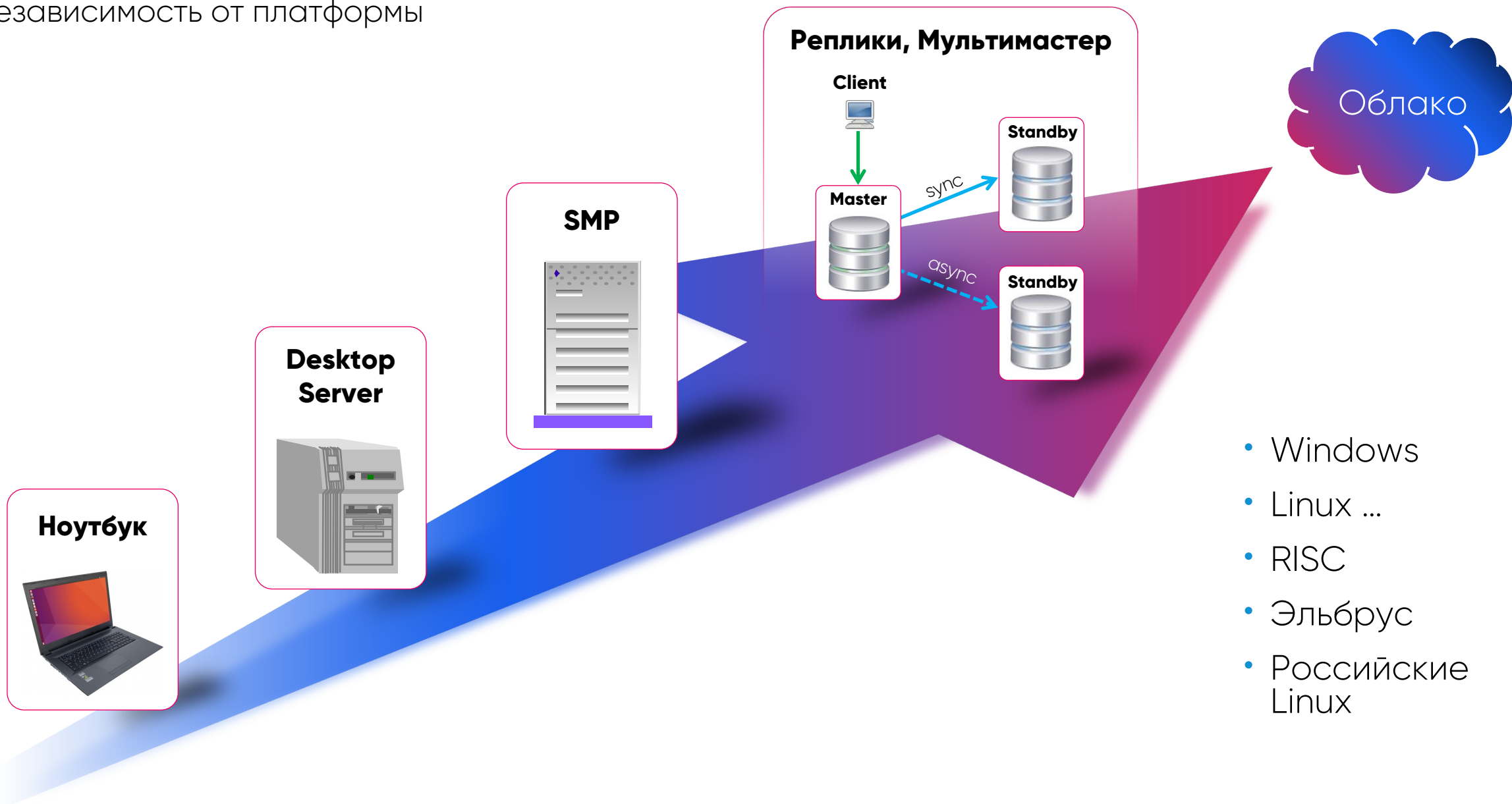
(версии 9, 10, 11, 12)



- **Многоплатформенность**
- **Расширяемость кода**
- **Различные типы данных**
- **Высокая надежность и доступность (24 x 7)**
- **Многосерверные архитектуры**
- **Безопасность**
- **Масштабируемость** (БД 150+ Tb, десятки тысяч пользователей, блокировки MVCC, 64 bit номер транзакции)
- **Производительность и эффективность** использования ресурсов
- **Управляемость**
- **Cloud** (DBaaS)

Разработав однажды, запускайте везде – 30+ платформ

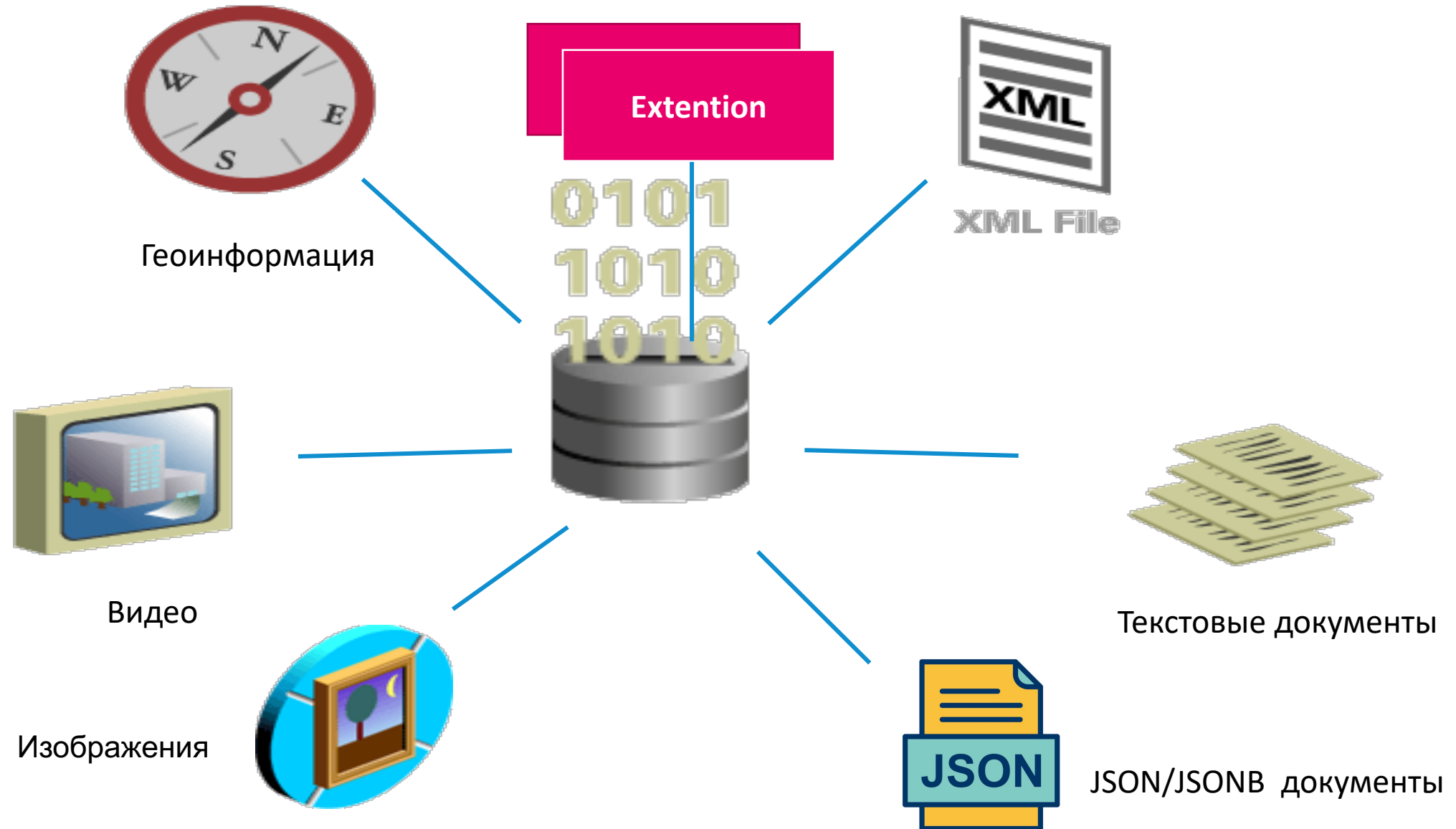
Независимость от платформы



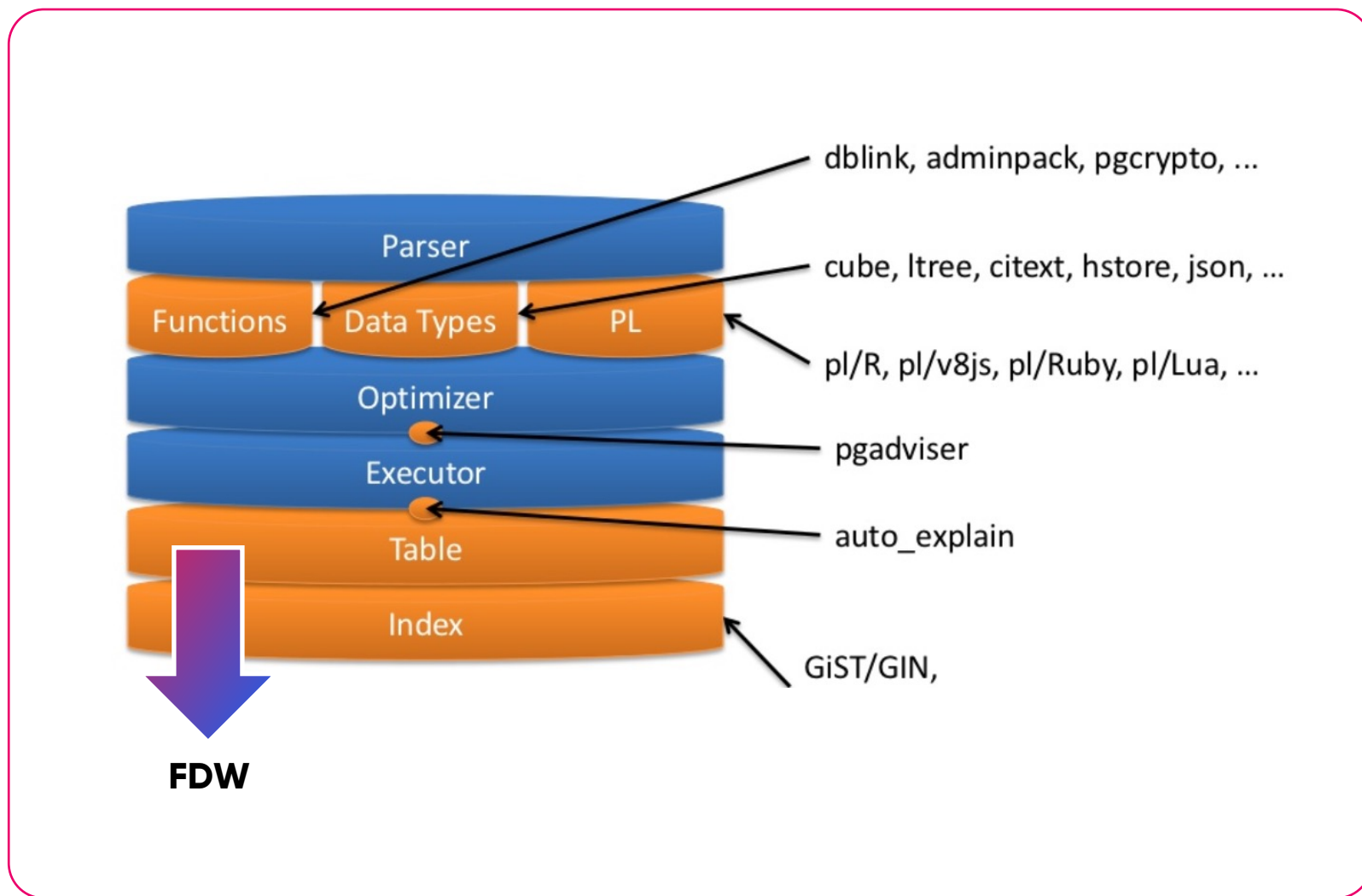
Различные типы данных, расширяемость



Работа с неструктурированными данными



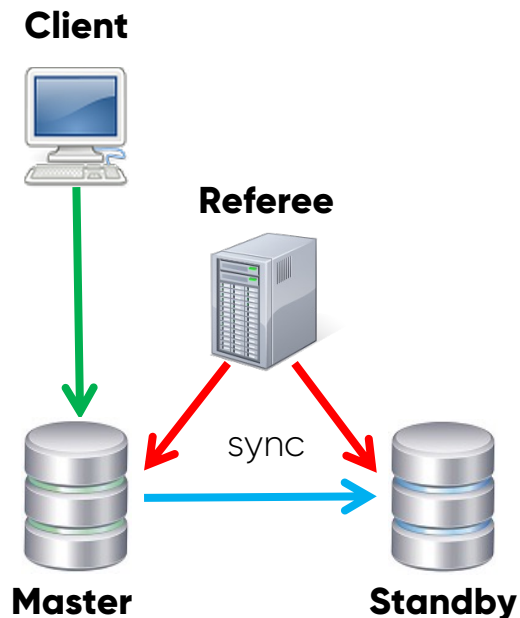
- **Extension** – фактически plugin для СУБД
- **Может добавлять новые типы данных, менять поведение**
- **Примеры:**
 - pg_partman (partitioning)
 - pg_variables (глобальные переменные)
 - pg_transfer (онлайн перемещение таблиц)



Высокая надежность и доступность

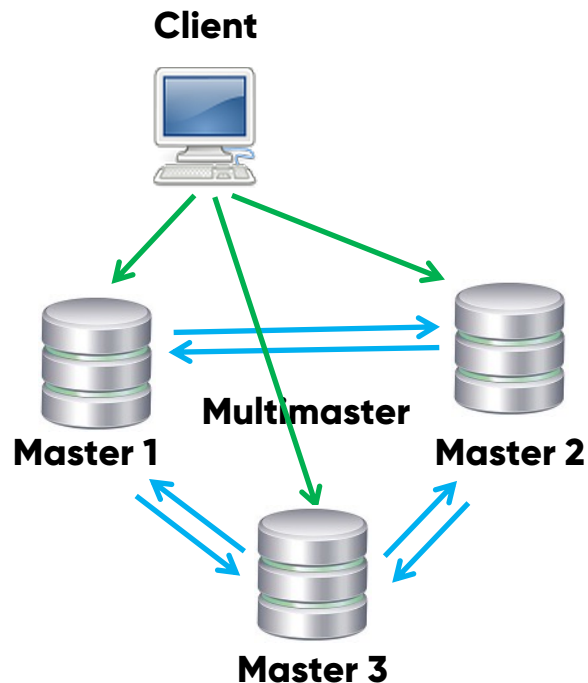
24 x 7, 5 сек задержки при
переключении узлов кластера





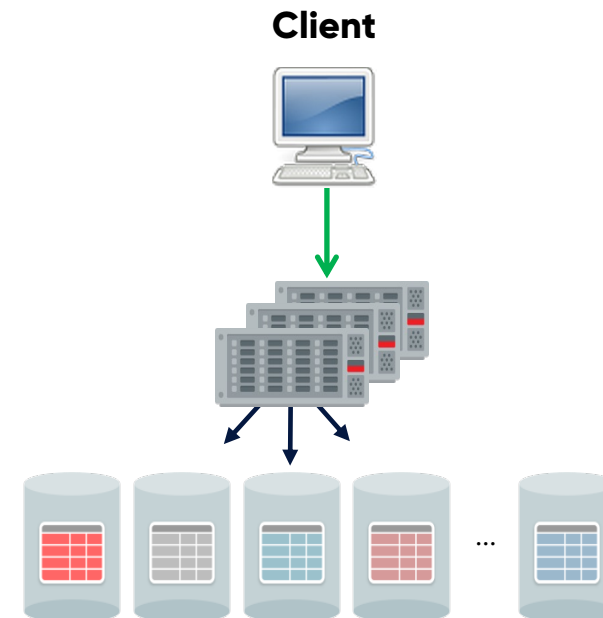
Реплика (Standby)

Катастрофоустойчивость,
надежность, масштабирование



Мультимастер

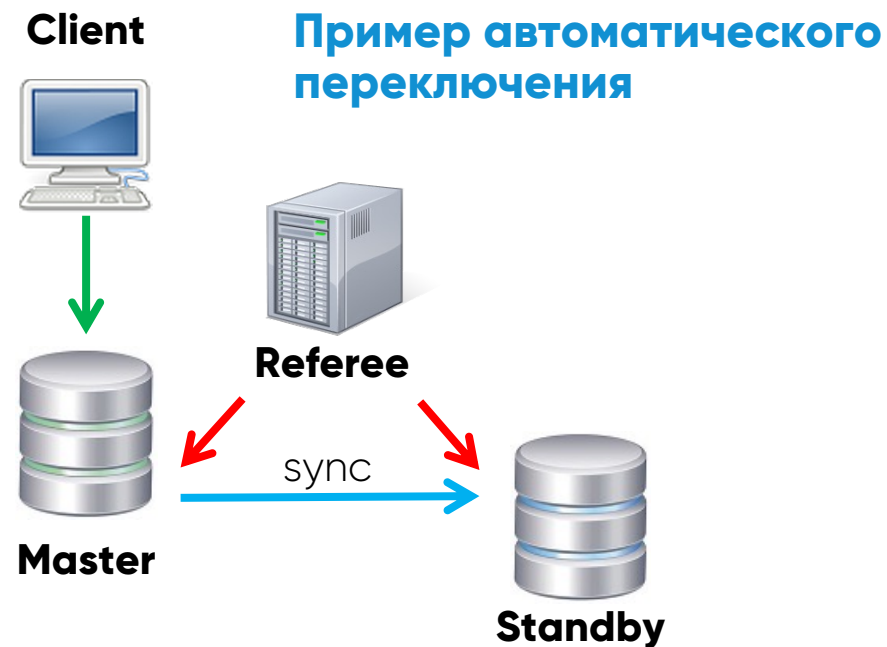
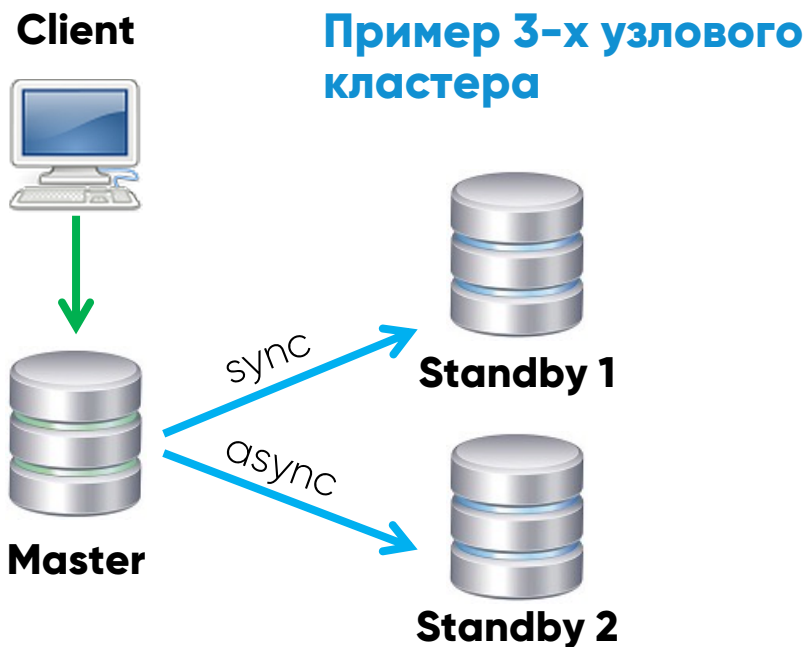
Быстрое переключение,
надежность, масштабирование



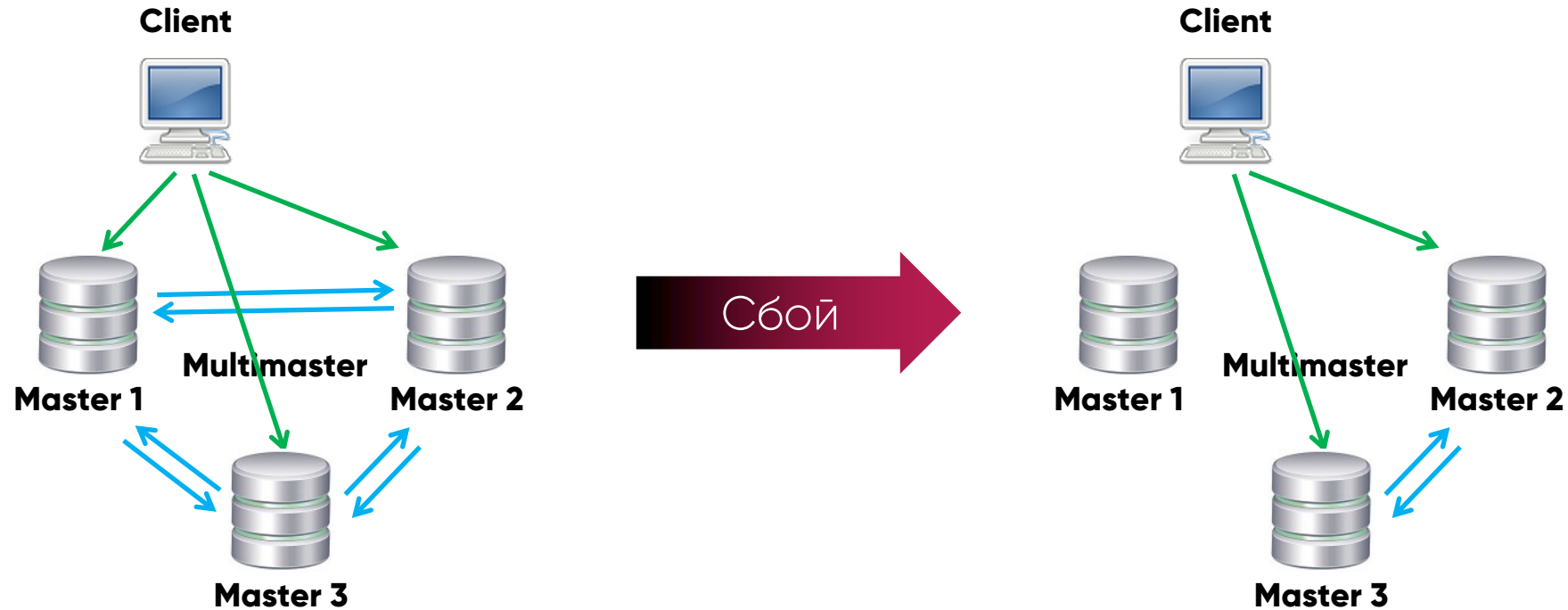
Шардинг

Снимает ограничения на размер
БД, масштабирование

Кластер на потоковой репликации (Standby)

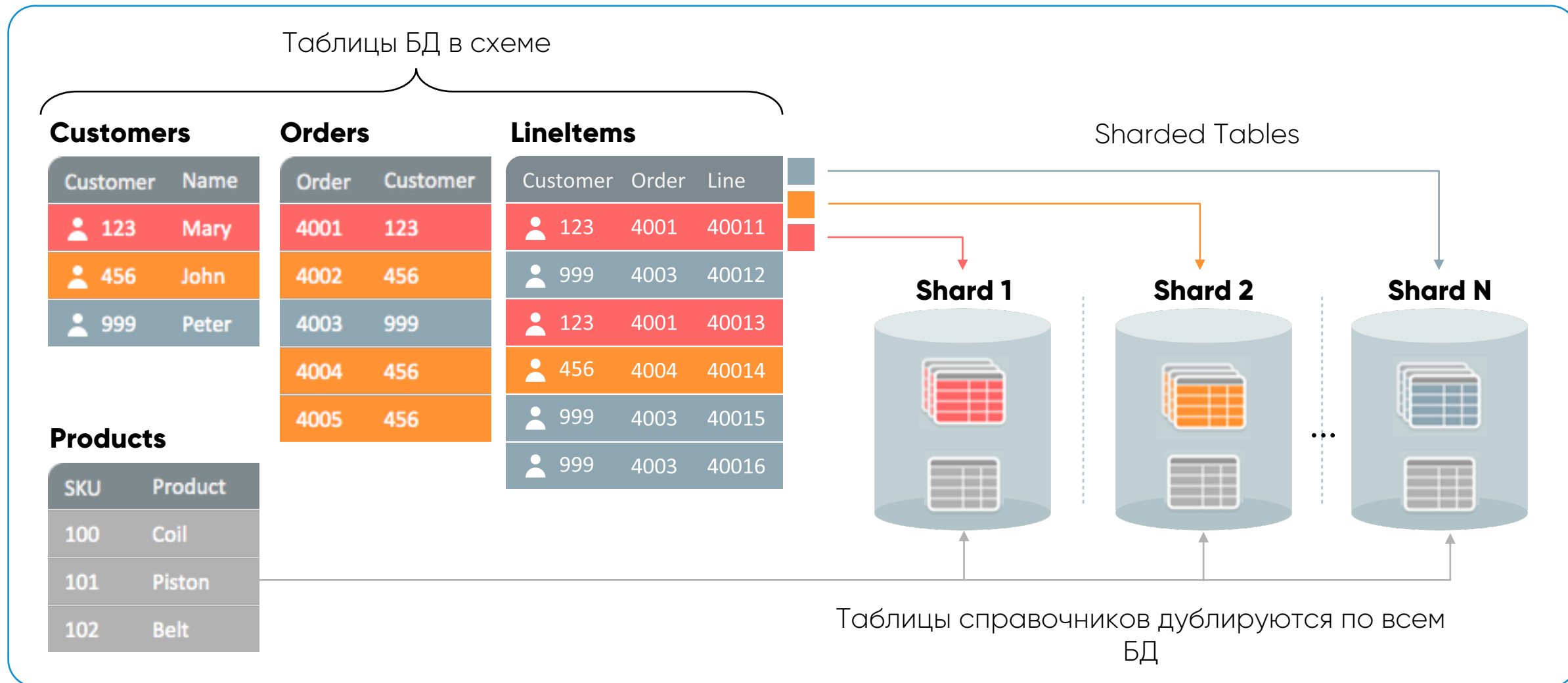


- Репликация синхронная/асинхронная, физическая/логическая
- Referee – Patroni, Stolon, Corosync
- Реплика (Standby) открыта на чтение – разгрузка мастера
- Небольшие оперативные in-memory таблицы открыты на запись
- Восстановление битых блоков из реплики, восстановление битых записей WAL
- Реплика может быть географически удалена

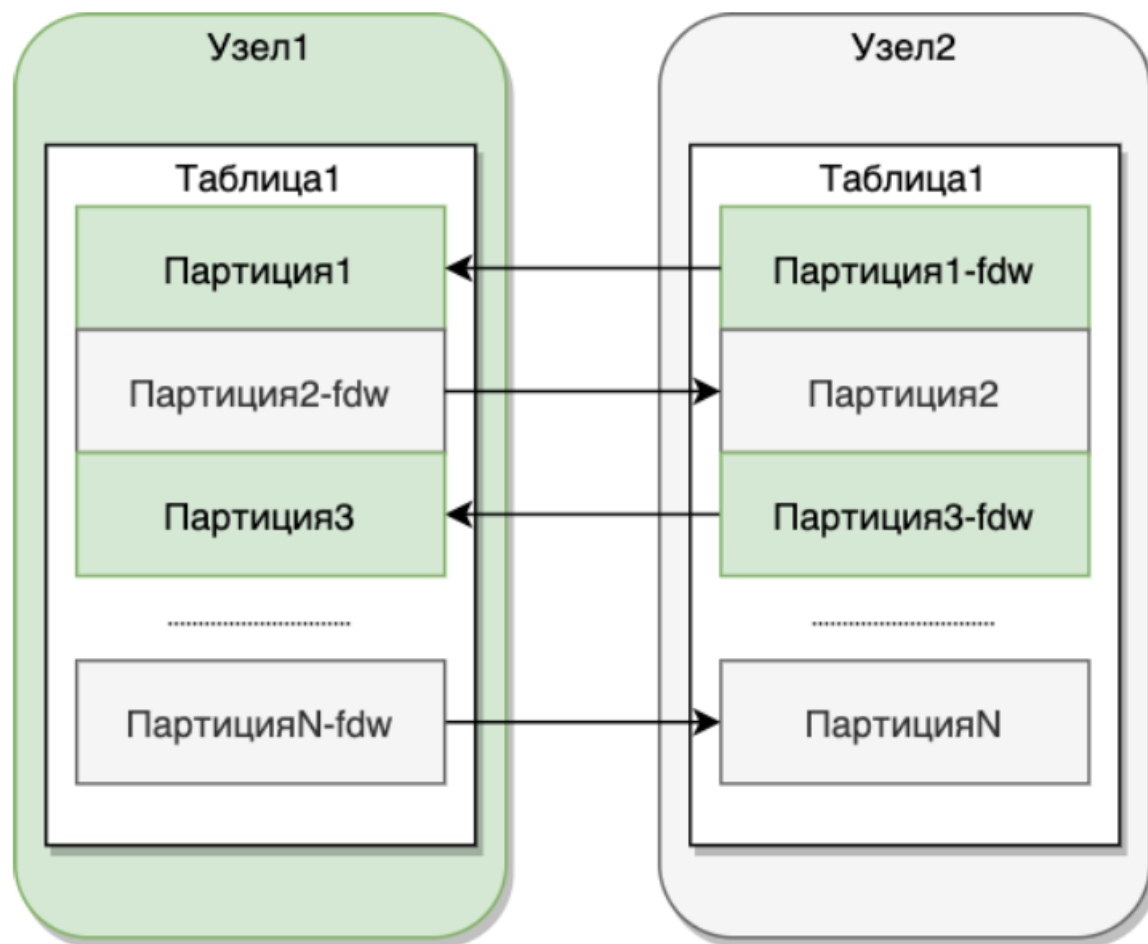


- Узлы открыты на чтение/запись
- 3 фазная фиксация изменений – замедляет работу
- Разрешение конфликтов
- Быстрое переключение

Шардирование по хэш функции снимает ограничения на размер БД



Шардман – масштабируемость и надежность



Распределенная таблица – это партицированная таблица, состоящая из N секций. Количество партиций указывается при создании распределенной таблицы. Части таблицы находятся на всех узлах кластера. Партиции связаны между собой через механизм `postgres_fdw`. Таблицы доступны на запись на всех узлах кластера. Каждый шард – отказоустойчив, в случае выхода одного из узлов шарда из строя, данные не будут потеряны.

Соразмещенная таблица – это партицированная таблица, состоящая из N секций, каждая секция которой соразмещена с распределенной таблицей.

Глобальная таблица – это таблица, размещенная на всех узлах кластера, содержащая идентичные данные.

- **Онлайн операции:**
 - Изменение параметров
 - Перестройка индексов
 - Изменение структуры таблиц
 - Работа с партициями
 - Патчи
- **Васкп/Восстановление БД, dump**
 - Инкрементальный (3 варианта) и полный Васкп, слияние полного и инкрементальных баскп
 - Параллельность баскп/восстановления
 - Сжатие баскп
 - Восстановление БД на момент времени (PITR)
 - Пробное восстановление
 - Политики хранения
 - [Васкп на объектное хранилище S3]
- **Dump – создание скриптов экспорта**
- **Автономные транзакции**

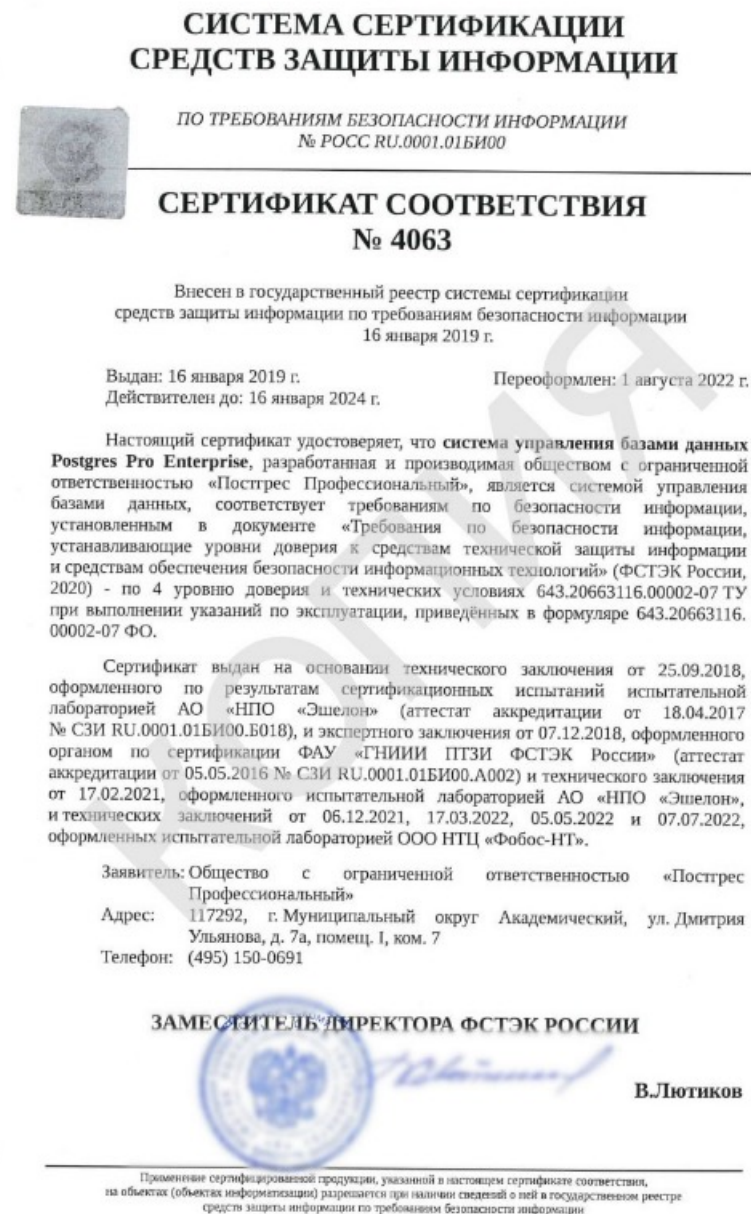
Информация всегда доступна



Безопасность



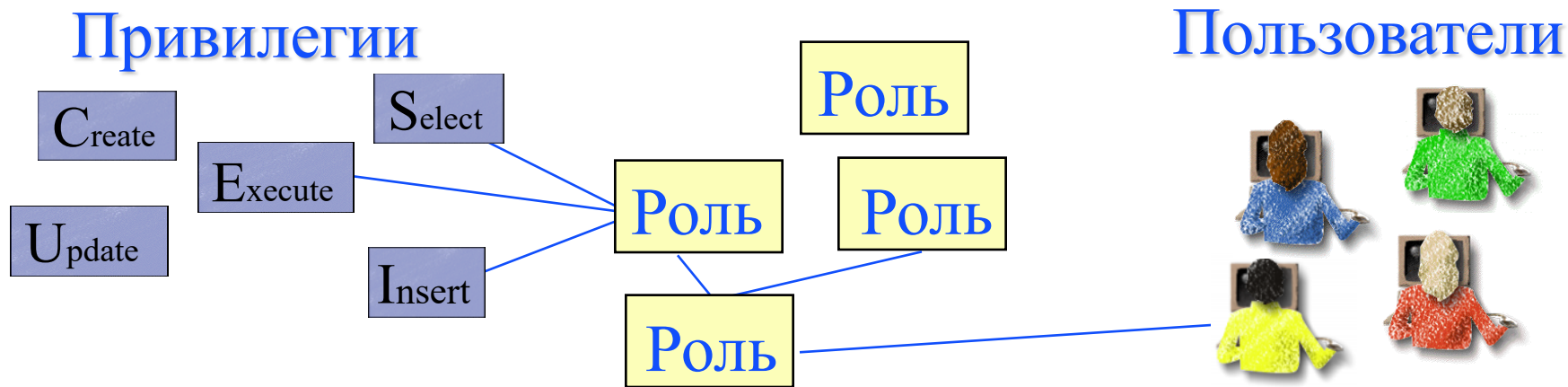
- **Сертификат ФСТЭК Г1**, уровень доверия УД 4 (защита конфиденциальной информации)
- У Oracle и Microsoft — **УД 6** (это хуже)
- **Проверенный российский код**
- **Российские разработчики и тех поддержка**
- **Очистка** выделенной оперативной и дисковой памяти
- **Coverity Scan Static Analysis**



Авторизация и аутентификация

- Имя/пароль
- Внешние системы
 - Kerberos
 - LDAP
 - Active Directory
 - Radius
 - PAM
 - peer/ident
- Авторизация ОС
- Авторизация БД
- Шифрование пароля
 - scram-sha-256
 - md5





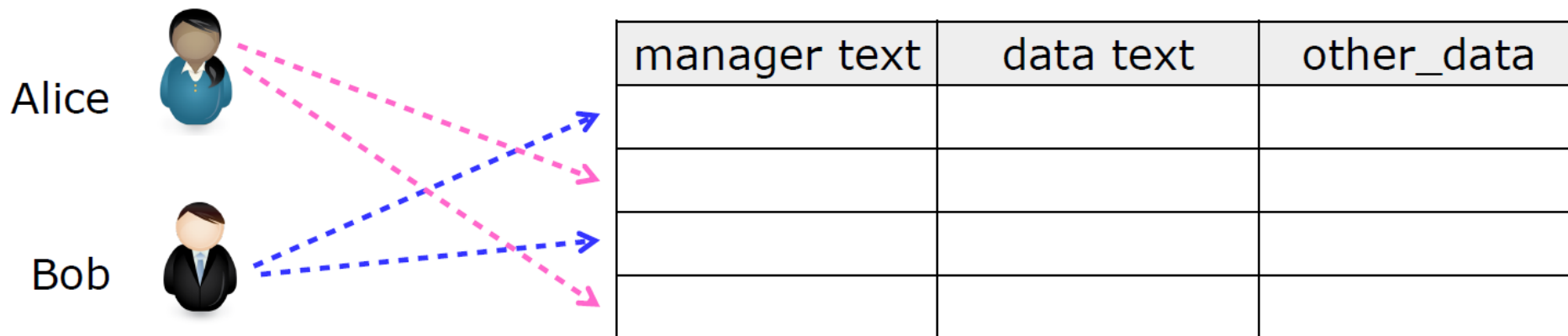
Объектные
Системные
Операционные
На таблицы и функции
От имени исполнителя/владельца
Superuser

Шифрование

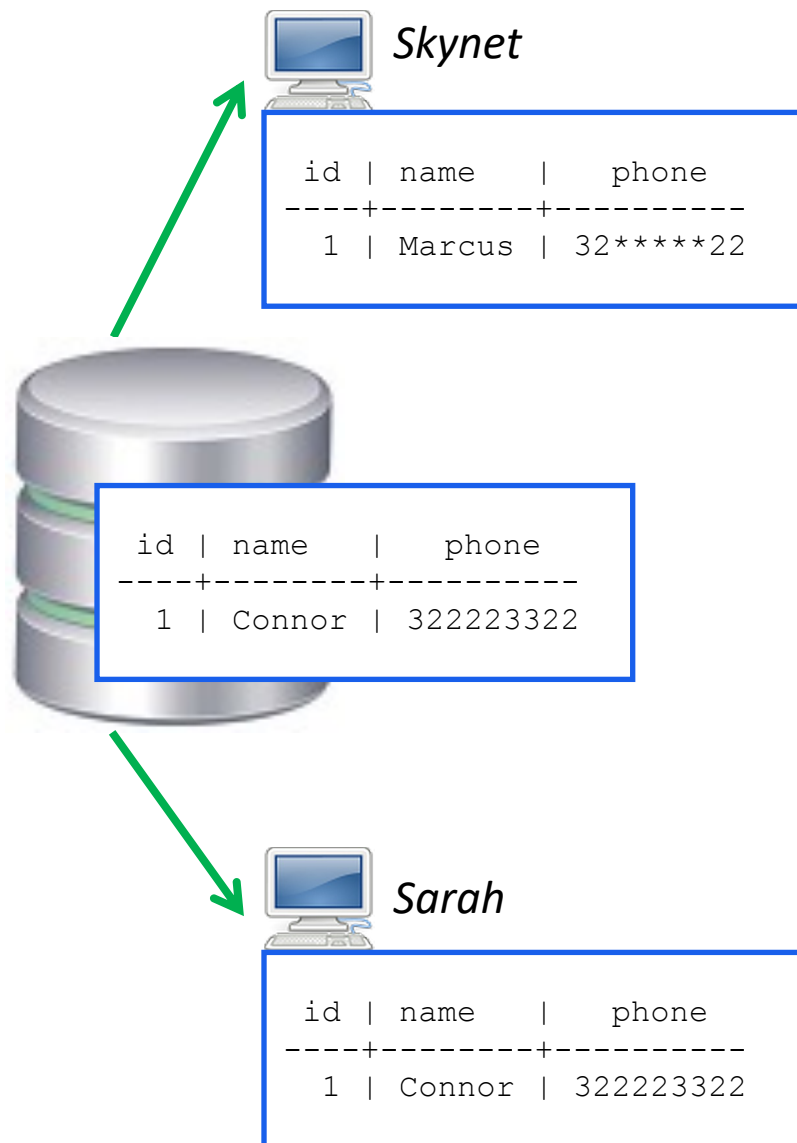
- Пароль
- Передача по сети - SSL
- Данные – с Алладин
- Российское – с Алладин

RLS – Row Level Security

```
ALTER TABLE accounts ENABLE ROW LEVEL SECURITY;  
CREATE POLICY manager_policy ON accounts USING(manager = current_user);  
SELECT * FROM accounts;
```



- **Поддерживается ответственным администратором...**
- **Гибкость и дискретность аудита**
 - Аудит пользователей, операций, обращений к объектам, привилегий ...
 - Множество событий, подлежащих аудиту
 - Уровни логирования в журнал: DEBUG5, DEBUG4, DEBUG3, DEBUG2, DEBUG1, INFO, NOTICE, WARNING, ERROR, LOG, FATAL и PANIC
- **Журналы:** stderr, csvlog, syslog, eventlog(wind), журнал аудита
- **pg_proaudit, Алладин**



pgpro_anonymizer —расширение для маскирования или замены конфиденциальных данных непосредственно внутри экземпляра Postgres Pro:

- Динамическое маскирование: сокрытие данных только от недоверенных пользователей.
- Статическое маскирование: преобразование данных в БД в соответствии с правилами.
- Экспортирование замаскированных данных в файл SQL.

Правила маскирования задаются метками безопасности:

```
SELECT anon.start_dynamic_masking();
```

```
SECURITY LABEL FOR anon ON COLUMN people.name  
IS 'MASKED WITH FUNCTION anon.fake_first_name();'
```

```
SECURITY LABEL FOR anon ON COLUMN people.phone  
IS 'MASKED WITH FUNCTION anon.partial(phone,2,$$*****$$,2);'
```

```
SECURITY LABEL FOR anon ON ROLE Skynet, Mark IS 'MASKED';
```

Масштабируемость

Использование всех ресурсов
компьютера, поддержка больших БД,
больших нагрузок

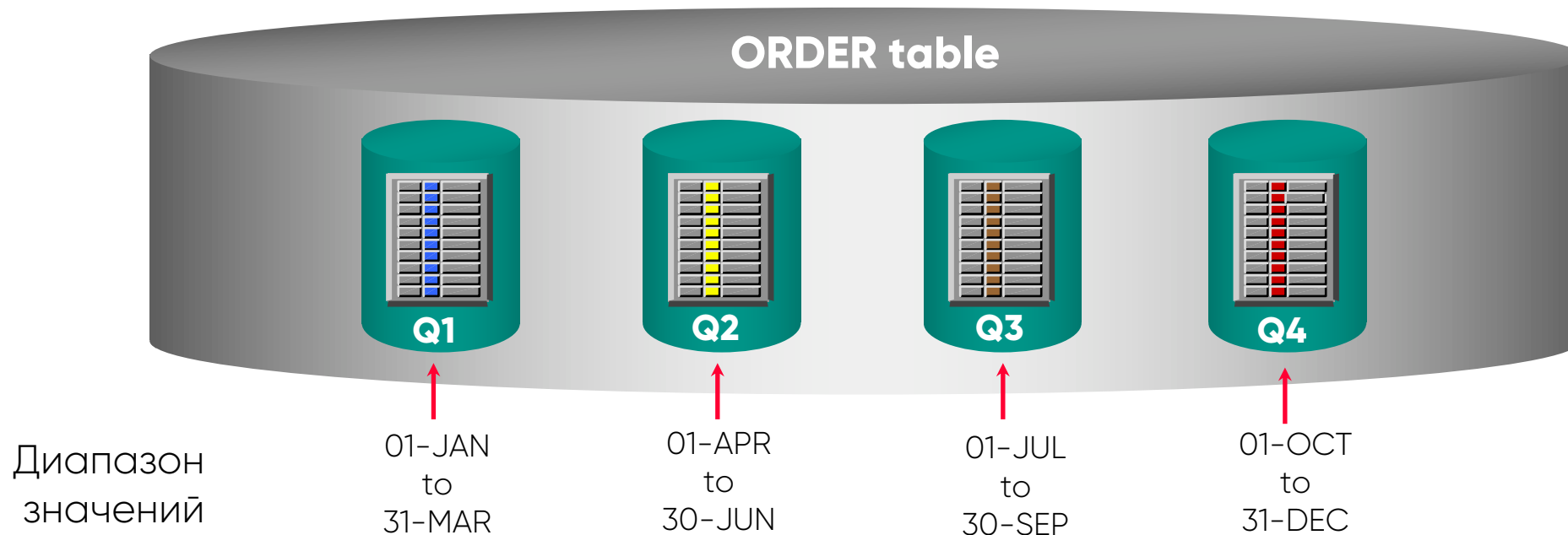


- **Разнесение нагрузки** – реплика, мультимастер, шардинг, встроенный load balancer
- **Секционирование**
- **Параллельная обработка**
- **Сжатие**
- **Управление ресурсами** (resource manager) – приоритизация для сеанса на использование `cpu`, `io_read`, `io_write` (1,2,4,8) 4-default, 8 – макс
- Сегодня у заказчиков есть БД 130+ Tb (Yandex mail – 300 Tb), поддержка 10000+ пользователей (1с – 4000 одновременных пользователей)

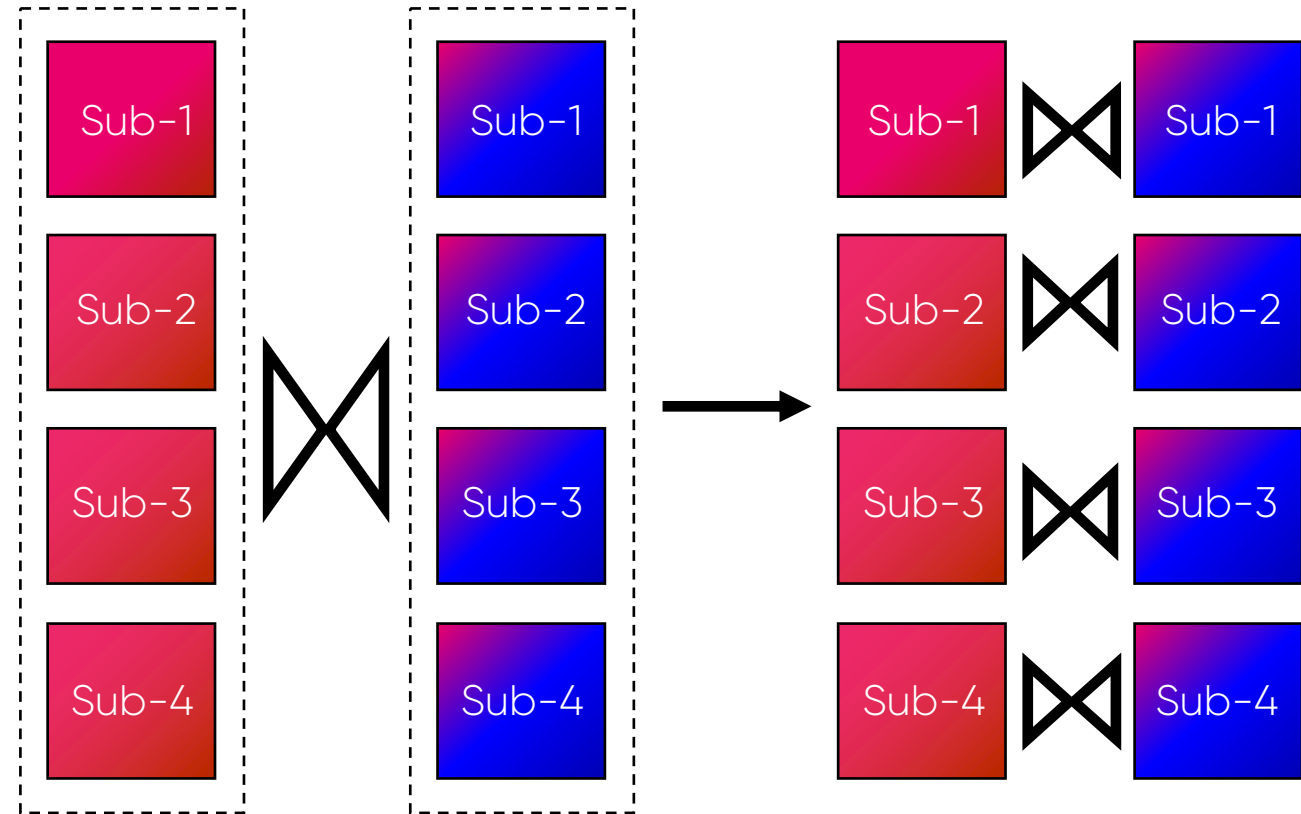
Секционирование таблиц и индексов

Улучшает доступность, управляемость

- Секционирование упрощает управление VLDB
- Секционирование значительно повышает производительность (partition wise)
- Разделы прозрачны для пользователей и приложений
- Данные разбиваются в зависимости от приложения
- Каждый раздел может иметь свой tablespace



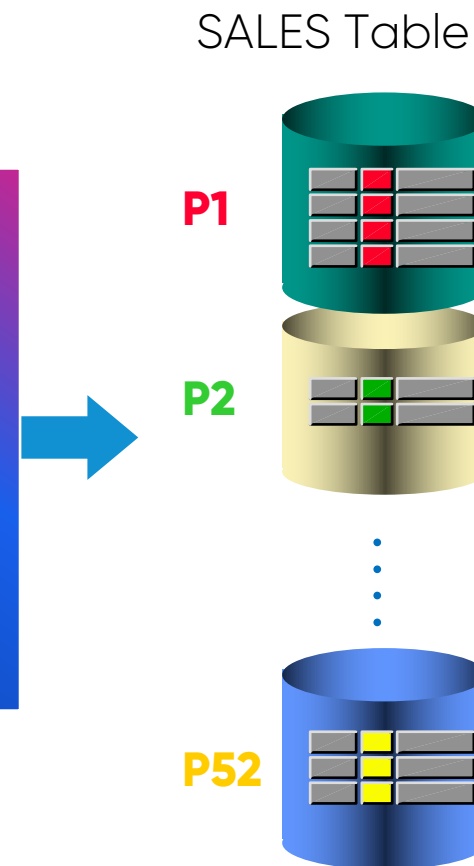
- **Hash** — по хэш функции
- **Range** — по диапазону
- **List** — по списку значений
- **Partition wise**
- **Автоматическое создание секций** (interval)
- **Композитное секционирование** (например, Range+Hash)
- **Онлайн операции над секциями**



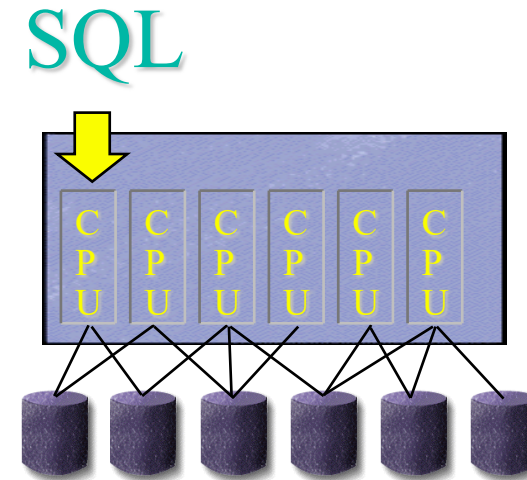
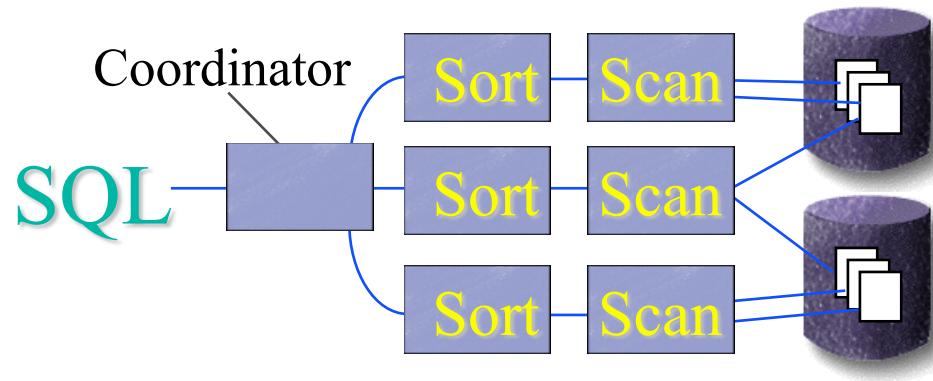
Секционирование по диапазону

- Строки запоминаются на **основе ключевого значения**
- Каждый раздел может иметь **собственный tablespace**
- Интеллектуальная **оптимизация запросов с ограничением** по частям таблиц
- **Параллельные операции** управления данными на частях таблиц
- **2 синтаксиса** —
pg_pathman (в примере) и
базовый (PostgreSQL)
(рекомендуется)

```
CREATE TABLE
sales (acct ..., rep ..., amt ..., weekno ...)
PARTITION BY RANGE(weekno) ...
(PARTITION P1 VALUE LESS THAN(4)
TABLESPACE data1,
PARTITION P2 VALUE LESS THAN(8)
TABLESPACE data2,
...
PARTITION P52 VALUE LESS THAN
(MAXVALUE)
TABLESPACE data52
```



Parallel Query



- max_parallel_workers_per_gather > 0
- Параллельное сканирование
- Параллельное сканирование индекса или по индексу
- Параллельное соединение, агрегирование
- Optimizer Hints (Администратор имеет контроль)
- Parallel-Aware Query Optimizer

- **Можно сжимать:**
 - Структурированные данные (таблицы, индексы)
 - TOAST таблицы
 - Неструктурированные данные (текст, XML и т д)
 - Backup – утилита pg_probackup
 - Dump – утилита pg_dump
- **CFS – Compression File System** –
встроена в PG Enterprise, задается
для Tablespace
- **Json → JsonB**
- **Выбор алгоритма и степени сжатия**
(lz4, zstd, zlib, pglz...)
- **Выгоды сжатия**
 - Уменьшает использование диска – обычно в 2 – 3 и до 5 раз
 - Уменьшает ввод/вывод



FDW – Foreign Data Wrapper

- **Секции и таблицы могут находиться вне текущей БД**

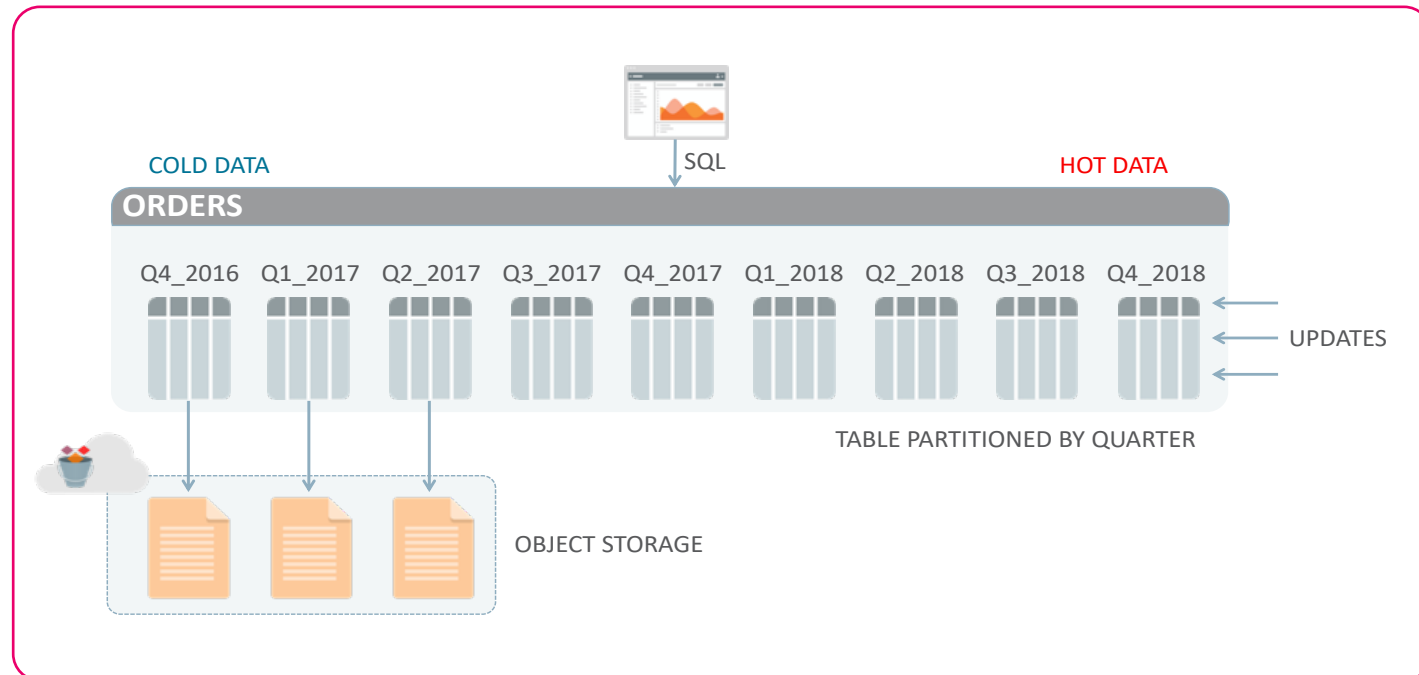
- В БД Postgres
- В других БД
- В файлах

- **Работа с FDW**

- Обеспечить доступ к SOURCE
- CREATE EXTENSION
- CREATE SERVER – ссылка на dbname, host, port)
- CREATE USER MAPPING (my_user → source_user)
- CREATE FOREIGN TABLE – ссылка на схему и таблицу с SERVER

- **Доступ через обертку FDW как к обычным объектам БД**

- Стандартные обертки
- Пользовательские обертки



- **Множество индексов, MV**
- **Инкрементальный backup**
- **MVCC – читатели не блокируют писателей, писатели – читателей**

- **Ограничения PG Enterprise**
 - Размер БД – не ограничен
 - Максимальный размер таблицы – 32 Tb
 - Количество записей в таблице – не ограничено
 - Количество индексов – не ограничено
 - Максимальный размер первичного ключа – 32 колонки
 - Максимальная длина записи – 400 Гб
 - Максимальная длина атрибута TOAST – 1 Гб
 - Максимальная длина BLOB – 4 Tb

Производительность



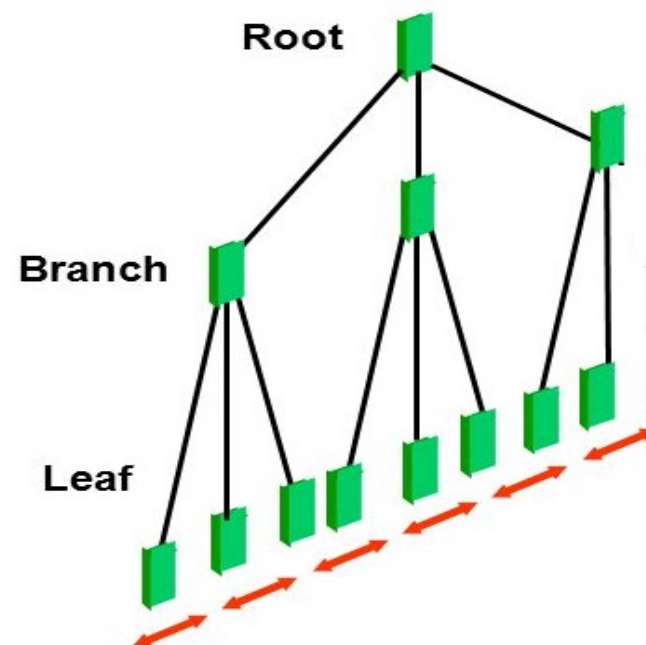
- Хороший оптимизатор (планировщик) запросов
- AQO – адаптивная оптимизации запросов (учет ошибок статистики при следующем выполнении запроса)
- Хинты + PG_hint_plan (SQL patch), заморозка планов
- Множество различных типов индексов
- Секции и материализованные представления, wise join
- Перенос табличных пространств на быстрые диски



Производительность

- **JSONB**
- **Prepared SQL** запросы
- **VACUUM ANALYZE** – сбор статистики для оптимизатора
- **64 – битный счетчик** транзакций
- Вынос нагрузки на **реплику, мультимастер, шарды**
- **JIT-компиляция**

- **В-деревья** ($>$, $\geq$, $<$, $\leq$, $=$)
- **Хеш** ($=$)
- **GIST** – (generalized search tree) – для **гео, текстов, рисунков** – разбивка на квадраты по ф-ии соответствия (близости, вхождению, пересечению, левее-правее, KNN и т д)
- **SP-GIST** – space partitioning GIST (разбиение на непересекающиеся квадраты)
- **GIN** – (Generalized Inverted Index – обратный индекс), например, **индекс в конце** книги – где встречается элемент). Хорош для **полнотекстового** поиска
- **RUM** – **развитие GIN**, поиск релевантных документов, **полнотекстовый** поиск, **знает ПОЗИЦИЮ** слова на странице, слова отстоящие на x слов, сортировка по похожести
- **BRIN** – (Block Range Index) – аналог **storage index** (исключение “блоков”)
- **Bloom** Index – на равенство, лучше В-деревя если надо индексировать много колонок таблицы. Он более компактен
- **Триграммы для GIST и GIN**
- Покрывающие и частичные индексы, индексы по выражению

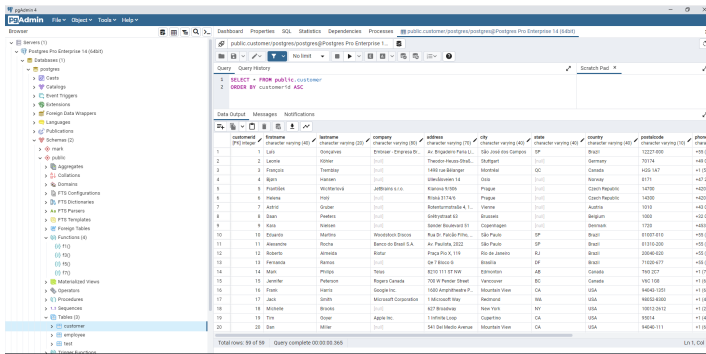


Управляемость

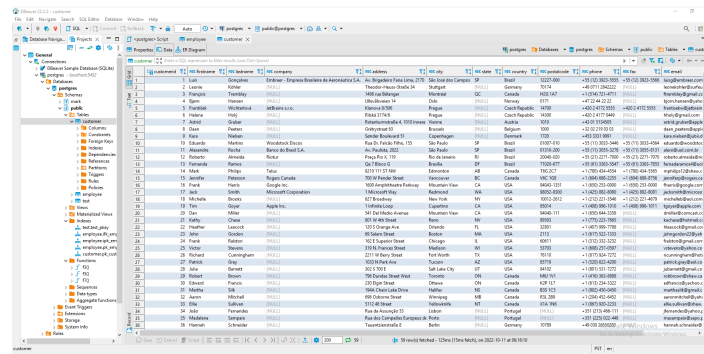


- **Управление объектами БД и кластера** (просматривать иерархию и характеристики и администрировать)
 - базы, TS, пользователи, схемы, роли, таблицы, MV.....
 - PgAdmin, Dbeaver, Valentina Studio, Navicat
- **Мониторинг поведения БД** (метрики, графики по времени, дашборды, богатые наборы шаблонов, тренды.....)
 - Zabbix+Mamonsu, Prometheus
- **Конфигурирование** (реплик, узлов, FDW, баз.....)
- **Диагностика** (поиск, выявление, предсказание проблем)
- **Настройка БД и запросов**

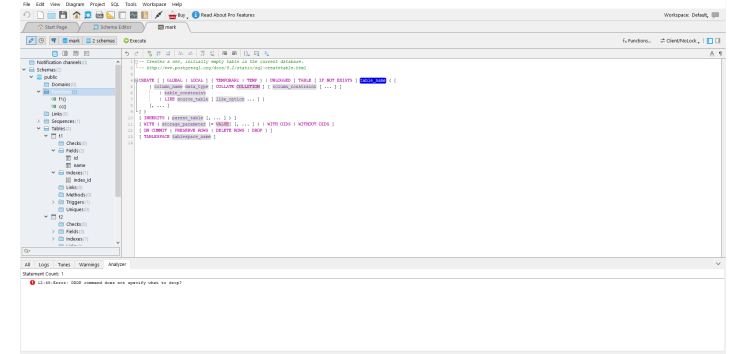
Графические инструменты



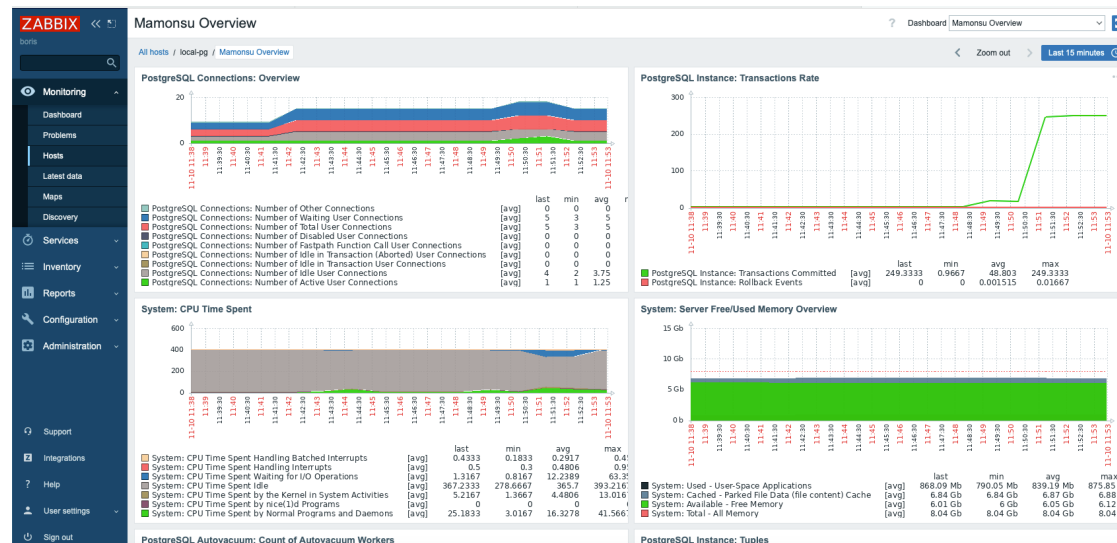
PgAdmin



Dbeaver



Valentina Studio



Zabbix + Mamonsu

Список БД в Postgres Pro Enterprise Manager – draft

PostgresPro

ENTERPRISE MANAGER

УПРАВЛЕНИЕ

Дашборд

Инстансы

Базы данных

Лог событий

Расписание

Кластеры

Бэкап

Профайлер

СИСТЕМА

Управление пользователями

Настройки

Аудит

TU Test User
Администратор агента, Адм...

Базы данных (4)
1002 MB

Имя БД

Размер

Wraparound

Инстанс

Таблицы

Индексы

pgbench

487.3 MB

∞

Instance on Agent

72 таблиц

420.35 MB

Bloat: 0.1%

123 индекса

67.4 MB

Bloat: 0.5%

pgpro_manager_repo

445.65 MB

∞

PgPro Manager repository

116 таблиц

363.87 MB

Bloat: 25.0%

216 индексов

81.77 MB

Bloat: 7.5%

postgres

34.2 MB

∞

PgPro Manager repository

77 таблиц

18.45 MB

Bloat: 2.1%

145 индексов

16.17 MB

Bloat: 1.5%

postgres

34.79 MB

∞

Instance on Agent

77 таблиц

18.61 MB

Bloat: 2.3%

145 индексов

16.6 MB

Bloat: 1.4%

- **Средства ОС** (ps, iostat, vmstat, sar, top, df, du, quota...)
- **Statistics Collector** – сбор статистики в БД (таблицы pg_stat_*), таблицы/view для отслеживания выполнения ANALYZE, CREATE INDEX, VACUUM, CLUSTER, COPY, backup, блокировок.
- **PGPRO_STATS** – мониторинг выполнения SQL
- **Журнал сообщений сервера** – ОС или утилита PGBADGER с GITHUB (only english)
- **EXECUTION PLAN**
 - EXPLAIN
 - EXPLAIN ANALYZE – реальное выполнение
- **Расширение PGPRO_PWR** (Pg workload reporting)

- [Top tables by blocks vacuum fetched](#)
- [Top indexes by blocks vacuum fetched](#)
- [Top tables by blocks vacuum read](#)
- [Top indexes by blocks vacuum read](#)
- [Top tables by WAL size generated by vacuum](#)
- [Top tables by vacuum operations](#)
- [Top tables by analyze operations](#)
- [Cluster settings during the report interval](#)

Server statistics

Database statistics

Database	Transactions			Block statistics			Block I/O times		Tuples					Temp files		Size	Growth
	Commits	Rollbacks	Deadlocks	Hit(%)	Read	Hit	Read	Write	Ret	Fet	Ins	Upd	Del	Size	Files		
db5	3377			100.00		185864			2129743	39349						9317 kB	
demo	17183			100.00		193323			2298257	43985						707 MB	
mamonsu_database	61593			100.00		359089			6689577	48937		1416		6998 MB	1337	9517 kB	
postgres	3764	5		99.93	1096	1554516	0.10		12364299	647192	21928	12584	14362	2306 kB	9	86 MB	4744 kB
Total	85917	5		99.95	1096	2292792	0.10		23481876	779463	21928	14000	14362	7000 MB	1346	811 MB	4744 kB

Session statistics by database

Database	Timings (s)			Sessions			
	Total	Active	Idle(T)	Established	Abandoned	Fatal	Killed
db5	80463.05	4.36	0.36	8			
demo	123699.00	225.80	0.33	11	3		
mamonsu_database	80411.25	192.43	0.34	8			
postgres	126511.75	10.96	4.11	25	1		

15

14

13

12

11

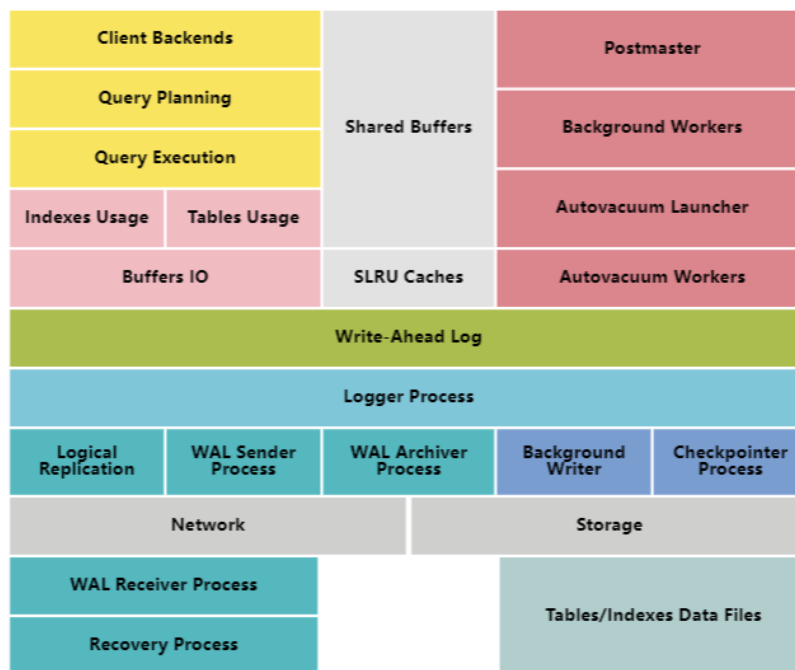
10

9.6

9.5

	<code>pg_stat_ssl</code>
	<code>pg_stat_activity</code>
<code>pg_backend_memory_contexts()</code>	<code>pg_backend_memory_contexts</code>
	<code>EXPLAIN</code>
<code>pg_stat_monitor</code>	<code>pg_stat_statements</code>
<code>pg_stat_kcache</code>	<code>pg_stat_user_functions</code>
<code>pg_wait_sampling</code>	<code>pg_prepared_xacts</code>
<code>pg_blocking_pids()</code>	<code>pg_locks</code>
<code>pg_stat_progress_cluster</code>	<code>pg_stat_progress_create_index</code>
<code>pg_stat_progress_copy</code>	<code>pg_stat_all_indexes</code>
	<code>pg_stat_all_tables</code>
	<code>pg_statio_all_indexes</code>
	<code>pg_statio_all_tables</code>
	<code>pg_statio_all_sequences</code>
<code>pg_get_wal_replay_pause_state()</code>	<code>pg_is_wal_replay_paused()</code>
	<code>pg_current_wal_lsn()</code>
	<code>pg_wal_lsn_diff()</code>
	<code>pg_ls_logdir()</code>
	<code>pg_current_logfile()</code>
<code>pg_ls_logicalmapdir()</code>	<code>pg_replication_slots</code>
<code>pg_ls_logicalsnapdir()</code>	<code>pg_stat_replication_slots</code>
<code>pg_ls_replslotdir()</code>	<code>pg_stat_replication</code>
<code>pg_stat_subscription_stats</code>	<code>pg_stat_subscription</code>
	<code>pg_stat_wal_receiver</code>
	<code>pg_stat_archiver</code>
	<code>pg_ls_archive_statusdir()</code>
	<code>pg_stat_database_conflicts</code>
	<code>pg_stat_recovery_prefetch</code>

Postgres Observability



<code>pg_buffercache</code>	
<code>pg_shmem_allocations</code>	
<code>pg_stat_slru</code>	
<code>pg_stat_activity</code>	
<code>pg_stat_database</code>	
<code>pg_stat_progress_vacuum</code>	
<code>pg_stat_progress_analyze</code>	
<code>pg_stat_all_tables</code>	
<code>pg_stat_wal</code>	
<code>pg_ls_waldir()</code>	
<code>pg_walfile_name()</code>	<code>pg_walfile_name_offset()</code>
<code>pg_current_wal_insert_lsn()</code>	<code>pg_current_wal_flush_lsn()</code>
<code>pg_last_wal_receive_lsn()</code>	
<code>pg_last_wal_replay_lsn()</code>	<code>pg_last_xact_replay_timestamp()</code>
<code>pg_stat_bgwriter</code>	
<code>pg_stat_progress_basebackup</code>	
<code>pgstattuple</code>	
<code>pg_tablespace_size()</code>	<code>pg_relation_filenode()</code>
<code>pg_database_size()</code>	<code>pg_relation_filepath()</code>
<code>pg_total_relation_size()</code>	<code>pg_filenode_relation()</code>
<code>pg_relation_size()</code>	
<code>pg_table_size()</code>	
<code>pg_indexes_size()</code>	
<code>pg_ls_tmpdir()</code>	
<code>pg_ls_dir()</code>	

nicstat

iostat

Прочее...



- **Расширяемый набор языков** для написания кода в БД
- **sql, pl/pgsql, pl/perl, pl/python, pl/tcl, pl/R, pl/java, ..., pl/v8 (JavaScript)**
- **Интерфейсы для создания клиентов:** Tcl, Perl, C, C++, PHP, Json, ODBC, JDBC, Embedded SQL in C, Python, Ruby, Java, ...
- **Множество сред разработки:** Dbeaver, Valentina Studio, Navicat....

СУБД Postgres Pro – первый в России коммерческий продукт на основе PostgreSQL. Входит в Единый реестр отечественных программ и баз данных Минкомсвязи.

Standard

Современная СУБД, включает все новые функции PostgreSQL и полезные доработки от компании

Enterprise

Наиболее полнофункциональная СУБД с высокой производительностью и масштабируемостью

Certified

Сертифицированные ФСТЭК версии Standard и Enterprise



* Специальная версия для 1c

Сравнение версий

Характеристика/ возможность

PostgreSQL

Postgres Pro Enterprise

Postgres Pro Enterprise Certified

Сертификат ФСТЭК

Обновление версии без
остановки сервера

Отказоустойчивая конфигурация:
мастер-мастер (мультимастер)

Встроенный пул соединений

64-битовый счетчик транзакций

Сжатие данных

Инкрементальное блочное
резервное копирование

Эффективное секционирование
(10-ки тысяч секций)

Автономные транзакции

Встроенный планировщик
заданий

Оптимизация запросов методами
машинного обучения - AQO

•

•

•

•

•

•

•

•

•

•

Сравнение версий

Характеристика/ возможность

PostgreSQL

Postgres Pro Enterprise

Шардман

•

JsonB

•

Мамонсу – агент мониторинга

•

PG_Probackup

•

PGPRO_PWR

•

Хинты

•

Покрывающие индексы

•

KNN: Индексный поиск
ближайших соседей

•

Комбинирование
полнотекстового индекса с
индексом релевантности (RUM)

•

Мониторинг
производительности и контроль
ожидания при выполнении
запросов

•

Характеристика/ возможность

PostgreSQL

Postgres Pro Enterprise

Триггеры событий входа

•

Приоритизация ресурсов

•

Online перемещение таблиц

•

Автоматическое исправление
страниц из реплики

•

Исправление WAL из буферов
памяти

•

Работа мастера при выходе из
строя одной из реплик

•

Изменение конфигурации
других сеансов

•

Отложенный сброс временных
таблиц на диск

•

Pg_wait_sampling – история и
профиль ожиданий

•

Режим автоподготовки
операторов

•

**Характеристика/
возможность**

PostgreSQL

Postgres Pro Enterprise

Пакеты (как в Oracle)

•

Orafce дополнительные системные пакеты

•

Ora2pg EE

•

Статическое маскирование

•

Динамическое маскирование

•

SR_PLAN – заморозка плана

•

SQL/JSON

•

Поддержка платформ ARM и Эльбрус

•

Backup в S3

•

Миграция на Postgres Pro



- **Много инструментов** Ora2Pg, sqlserver2pgsql, Ispirer, Pentaho kettle, Diasoft
- **Расширение OraFCE**
- **Множество проектов** сделали партнеры
- **Миграция структуры и данных** – автоматическая
- **Миграция кода** – полуавтоматическая
- Это не **"ROCKET SCIENCE"**
- У Ora2PG – **оценка трудоемкости и продолжительности миграции**
- **Консультации, тренинг, HandBook, White Paper**
- **При миграции обязательны тестирование** (функциональное и нагрузочное), построение **архитектуры надежности, безопасности** и т д



PostgresPro

Q&A

