



Postgres Pro Enterprise 16

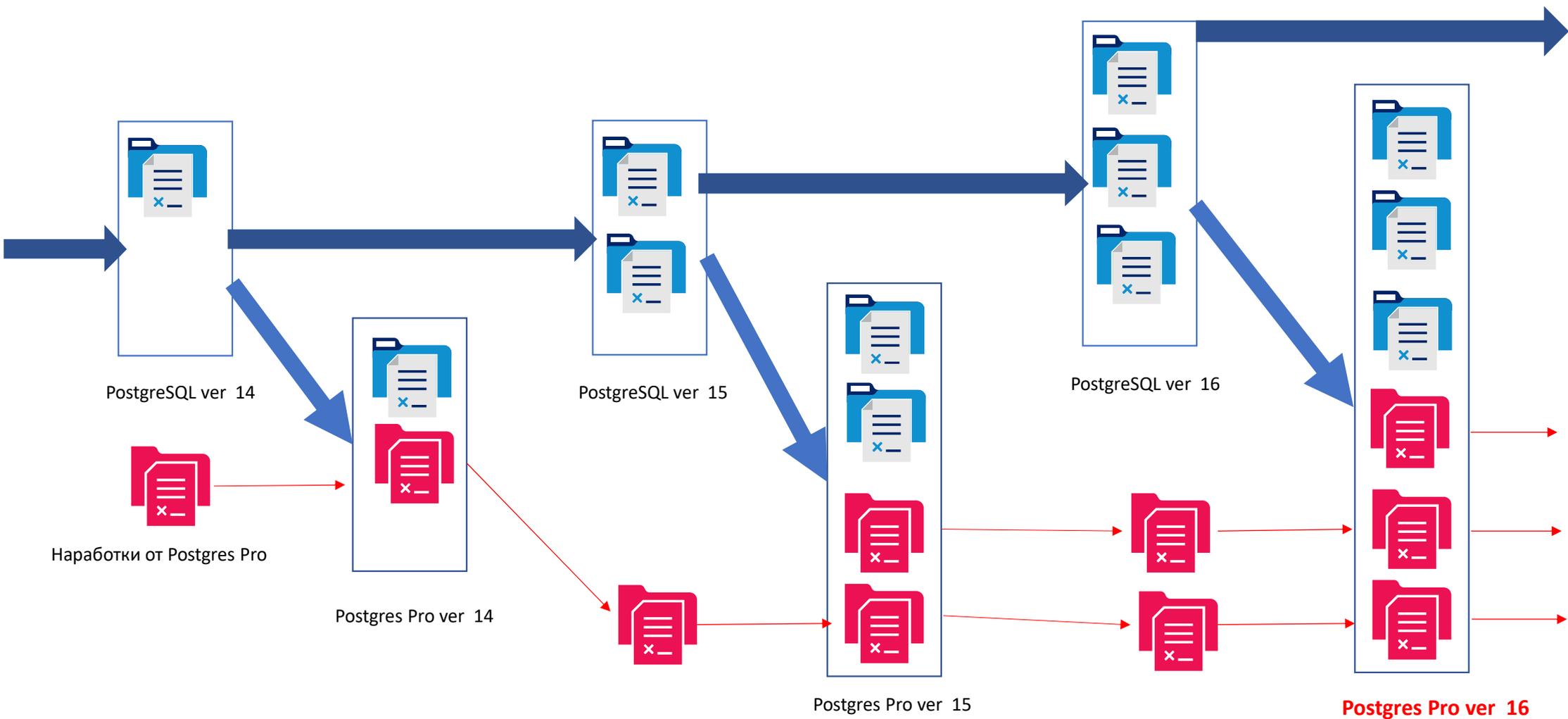
НОВЫЕ ВОЗМОЖНОСТИ

Марк Ривкин

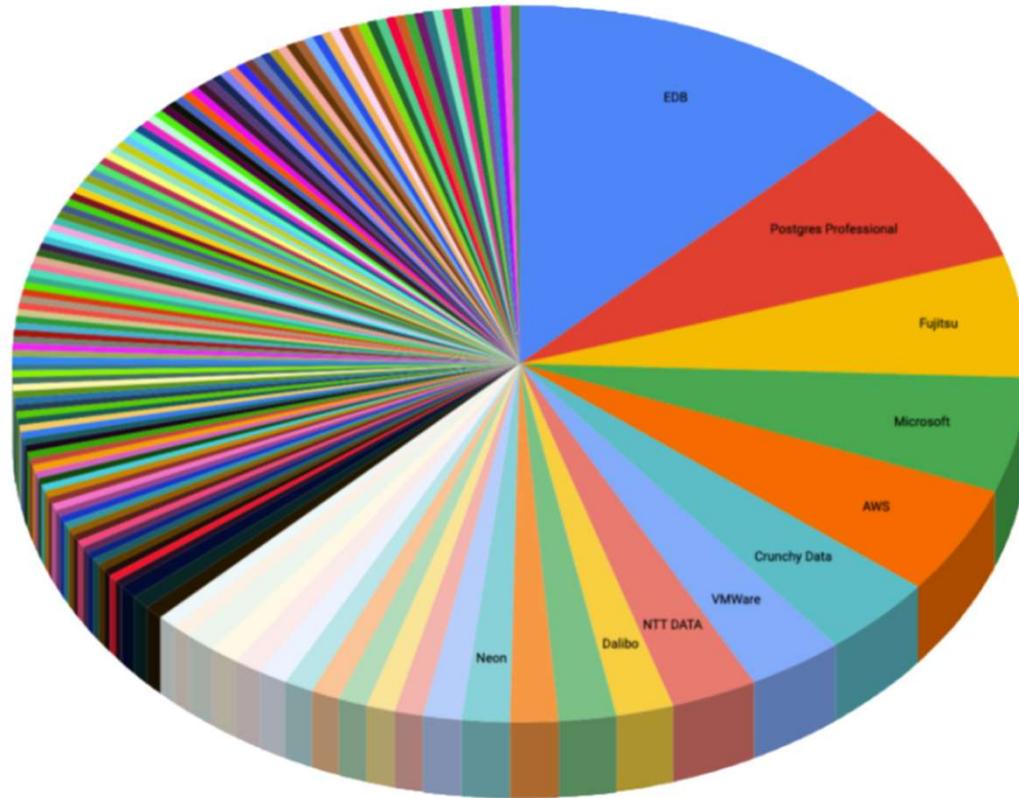
m.rivkin@postgrespro.ru

Слияние версий PostgreSQL и Postgres Pro Enterprise

(версии 14, 15, 16, 17)



Отчет EDB о контрибьюторах в PostgreSQL 15



Code contributors to Postgres by company (PostgreSQL.org + PG 15 Release Notes) without personal or freelance

- EDB
- Postgres Professional
- Fujitsu
- Microsoft
- AWS
- Crunchy Data
- VMWare
- NTT DATA
- Dalibo
- Timescale
- HighGo
- Neon
- Cybertec
- Adjust
- Credativ
- Google
- Kontur
- Materialize
- NTT
- Red Hat
- SRA OSS
- Yugabyte
- Aiven
- Instaclustr
- Loxodata
- pganalyze
- PostgreSQL Experts
- AD Parts
- Analytics Engines
- Anastigmatix
- Apple
- Arcion Labs
- Arenadata
- Atos
- Avaya
- Axians NL
- Bank of China
- BCL
- Betsys
- Betterment
- Bigbank
- Blacksmith Applications
- Braintree
- Caesars Digital
- Capital Rx
- Capsico Health
- CdC
- Citus Data
- Clearco
- Cockroach Labs
- Code Synthesis Tools
- Codice Lieve
- Cofano Software Solutions
- Coinbase
- Conova Communications GmbH
- CrateDB
- CRSCube
- Data Egret
- dbi services
- Deutsche Telecom
- Dext
- DockYard
- Doctolib
- DuckDB Labs
- EdgeDB
- End Point Corp
- Entelect
- EPAM Systems
- Fivetran
- Forest Management Institute
- Garner
- Gentoo
- GLS Bank
- GTT
- HeteroDB
- HP
- IBM
- ILande
- Illuminated Computing
- Index
- Instructure
- Intel
- Intellisoft
- Intezer
- JackDB
- Jampp
- 58 more

Источники новых возможностей

16.1

- PostgreSQL 16
- Postgres Pro Enterprise 15
- Разработки 2023 года
- Интеграция популярных расширений
- Плашка 16.1

Новые возможности 16 релиза СУБД Postgres Pro Enterprise

PEEM - Enterprise Manager

DBaaS

High Availability

Встроенный HA кластер BiNA

Интеграция Pg_probackup с другими СРК, pf_probackup 3.0

Миграция с Oracle

Пакеты 2.0

Системные пакеты (dbms_application_info, utl_mail, utl_http, utl_smtp)

Преобразование работы с коллекциями (методы массивов и pl/sql таблиц) в объекты PG (СУБД, Ora2pgpro, pg_variables)

Ассоциативные (ключ-значение) таблицы в Pg/PLSQL (аналог PL/SQL таблиц)

Bfiles

DBMS_LOB

Temporary LOBs

Производительность

Resource manager (управление и приоритизация ресурсов)

Адаптивный экзекутор (перепланирование запроса «на лету»), AQO 2.0, sr_plan

Baseline (управление планами запросов)

Шардман (OLTP)

CITUS (Analytics)

Безопасность

DVA без доступа к данным

Transparent Data Encryption

Новые требования ФСТЭК

Разработка

Pluggable TOAST, суперфайлы

PostgresPro Enterprise Manager – PPEM



- Графическое приложение для мониторинга и **управления** БД Postgres – для ДБА
- Концепция «единого окна» - все БД в одной консоли
- Визуализация основных метрик производительности - графики с выбором интервала наблюдения, статистику по SQL
- Можно посмотреть активность и ожидания по сессиям, журнал событий
- Запустить/остановить/перезапустить инстанс
- Бэкап и восстановление БД
- Изменить параметры настройки
- Вывести общую информацию – Dashboard
- Доступ к консоли командной строки PSQL прямо из браузера !

Dashboard – сводная информационная панель



PostgresPro
ENTERPRISE MANAGER

УПРАВЛЕНИЕ

- Дашборд
- Инстансы
- Базы данных
- Лог событий
- Расписание
- Кластеры
- Бэкап
- Профайлер

СИСТЕМА

- Управление пользователями
- Настройки
- Аудит

PSQL

Дашборд

ИНСТАНСЫ **2**

СЕРВЕРЫ **2**

КЛАСТЕРЫ **0**

ПОСЛЕДНИЕ УВЕДОМЛЕНИЯ

Похоже здесь пусто

БАЗЫ ДАННЫХ **4**

ОБЩИЙ РАЗМЕР БД **1002 MB**

ЛОГ СОБЫТИЙ

Дата, время	Инстанс	База данных	Тип	Log
12.04.2023, 11:47:04.43	PgPro Manager repository	pgpro_manager_repo	LOG	duration: 555.846 ms rows: 1 size: 7 bytes statement: WITH cfs_write_speed_lates FROM monitoring_history mh JOIN metrics m ON mh.metric_id = m.id WHERE m.metric_value_set DESC LIMIT 1), cfs_scan_speed_latest AS (SELECT mh.metric_value FF
12.04.2023, 11:47:03.16	PgPro Manager repository	pgpro_manager_repo	LOG	duration: 554.918 ms rows: 1 size: 7 bytes statement: WITH cfs_write_speed_lates FROM monitoring_history mh JOIN metrics m ON mh.metric_id = m.id WHERE m.metric_value_set DESC LIMIT 1), cfs_scan_speed_latest AS (SELECT mh.metric_value FF
12.04.2023, 11:46:33.38	PgPro Manager repository	pgpro_manager_repo	LOG	duration: 1076.063 ms rows: 105 size: 15074 bytes statement: SELECT m.metric_key mh.metric_value, m.datatype, m.units, m.delta, mh.value_set::timestamp(0), EXTRA metrics m JOIN monitoring_history mh ON (m.id = mh.metric_id) WHERE lower(m.metri mh.value_set >= '2023-04-12 08:31:28' AND mh.value_set <= '2023-04-12...
12.04.2023, 11:46:33.34	PgPro Manager repository	pgpro_manager_repo	LOG	duration: 1463.161 ms rows: 105 size: 14580 bytes statement: SELECT m.metric_key mh.metric_value, m.datatype, m.units, m.delta, mh.value_set::timestamp(0), EXTRA metrics m JOIN monitoring_history mh ON (m.id = mh.metric_id) WHERE lower(m.metri mh.value_set >= '2023-04-12 08:31:28' AND mh.value_set <= '2023-04-12...

Список инстансов

PostgresPro
ENTERPRISE MANAGER

УПРАВЛЕНИЕ

- Дашборд
- Инстансы**
- Базы данных
- Лог событий
- Кластеры
- Бэкап
- Профайлер

СИСТЕМА

- Управление пользователями
- Настройки
- Аудит

Инстансы

5 инстансов | 3 сервера | 0 кластеров

ДОБАВИТЬ ИНСТАНС

Наименование	Сервер	Чексуммы	Сбор логов	Роль	БД	
14/main Порт: 5432 Директория: /usr/lib/postgresql/14/bin Версия Postgres: 14.7	619DDD95962F agent-postgres Запущен	off	<input type="checkbox"/>	primary	Базы данных: 2 Транзакций в секунду: 1.03 Соединения: 1 Средняя нагрузка: 0.10 / 0.15 / 0.16	
PgPro Manager repository Порт: 5432 Директория: /opt/pgpro/ent-14/bin/ Версия Postgres: 14.6 (enterprise)	BAFA824CE87C 172.26.0.2 Запущен	on	<input checked="" type="checkbox"/>	primary	Базы данных: 2 Транзакций в секунду: 12.63 Соединения: 1 Средняя нагрузка: 0.10 / 0.15 / 0.16	
PgPro-EE Порт: 5432 Директория: /opt/pgpro/ent-14/bin Версия Postgres: unknown	N/A agent-pgproeeqqq Приостановлен	-	<input checked="" type="checkbox"/>	n/a	Базы данных: - Транзакций в секунду: - Соединения: - Средняя нагрузка: - / - / -	
rew Порт: 5432 Директория: / Версия Postgres: -	N/A agent-postgres Приостановлен	-	<input type="checkbox"/>	n/a	Базы данных: - Транзакций в секунду: - Соединения: - Средняя нагрузка: - / - / -	
test Порт: 5432 Директория: / Версия Postgres: unknown	N/A agent-pgproeeqqq Приостановлен	-	<input type="checkbox"/>	n/a	Базы данных: - Транзакций в секунду: - Соединения: - Средняя нагрузка: - / - / -	

Графики производительности



PostgresPro ENTERPRISE MANAGER

← НАЗАД

ДЕТАЛИ ИНСТАНСА

Метрики

Активность

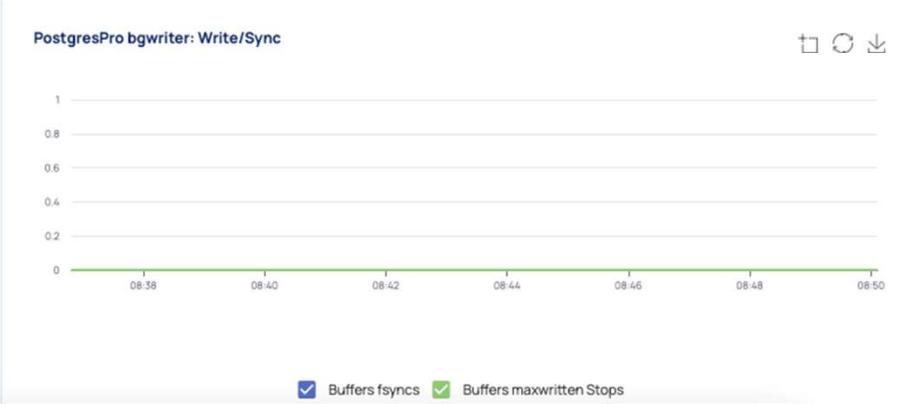
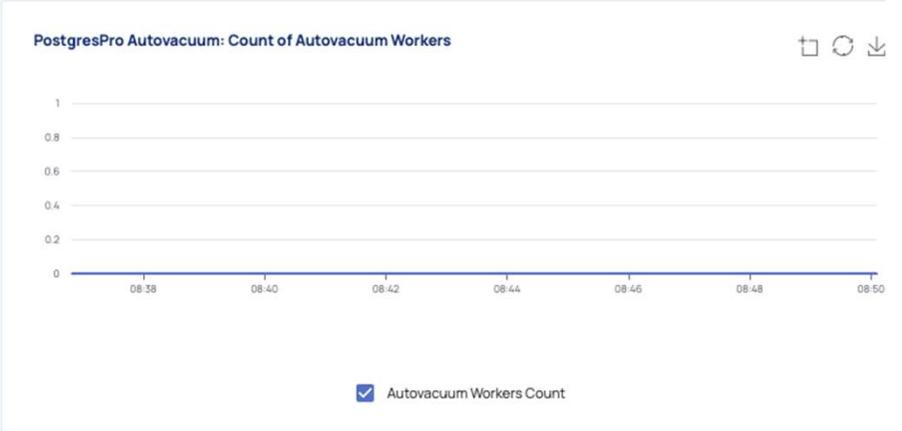
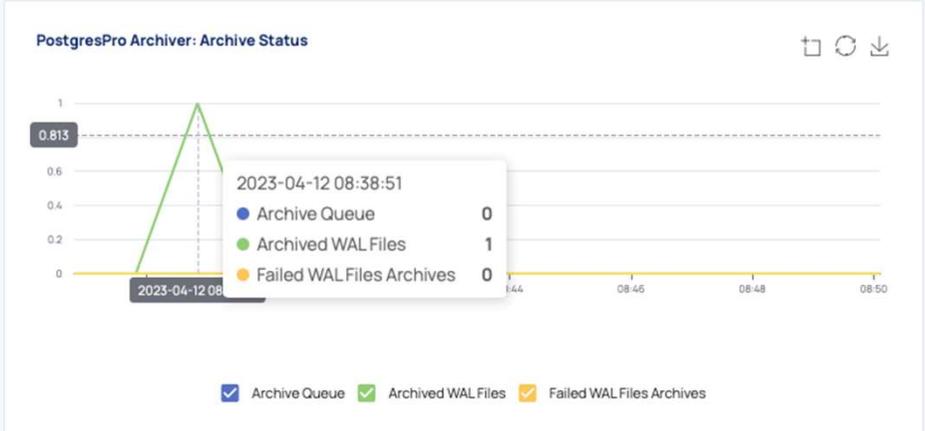
Статистика

Бенчмарк

Настройки

PSQL

ГРАФИКИ (24) Период: Последние 15 минут СОЗДАТЬ ГРАФИК



Версия 1.1 (сентябрь)

- Показ плана активного запроса (`pg_query_state`)
- Пользовательский отчет по составу ролей
- Выбор группы метрик в графиках (Блокировки/Производительность/Рабочая нагрузка/Фоновые процессы/CFS)
- Система тегирования Инстансов
- Progress Bar для бекапов и валидации
- Управление `pg_hba.conf` (сеть, доступ, шифрование пароля)
- Настройка вида таблиц отображения информации (какие колонки, порядок и т д)
- Подсветка синтаксиса SQL
- Поиск объектов по системе (Таблицы, Инстансы, Базы, и тп)
- Отображение/скрытие разделов в зависимости от привилегий пользователя

Далее (после релиза 1.1)

- Управление
 - Создание БД для 1С
 - PG_PWR
 - LDAP
 - Multimaster
 - ВИА
 - Shardman
 - Tablespace и CFS
 - Ora2Pgpro
- Просмотр и редактирование содержимого БД, функций, процедур, пакетов.
- Просмотр и изменение свойств объектов БД и доступа к ним.

DBaaS

- СУБД и БД в облаке
- Для конечного пользователя, которому быстро нужна БД
- Не просто БД в виртуальной машине, а
 - Портал самообслуживания
 - Возможность заказать БД и управлять ею.
 - Оплата по факту использования/учет использования ресурсов
- Веб-интерфейс портала самообслуживания DBaaS позволяет :
 - Создать инстанс кластера Postgres Pro, в том числе отказоустойчивый кластер с одним ведущим и несколькими резервными инстансами (при выходе из строя ведущего автоматически новым ведущим становится синхронный резервный инстанс)
 - Выполнять резервное копирование (pg_probackup) и создавать логическую копию (pg_dump) баз данных
 - Восстанавливать БД из бэкапа, переключаться на реплику
 - Администраторам DBaaS вести список пользователей и назначать для них квоты на потребление дискового пространства и использования ресурсов CPU и RAM
 - Сейчас на базе облака VK, будет Open Stack (для on-prem)
 - Ручной минорный апгрейд

DBaaS

PostgresPro
CLOUD MANAGER

user1
4982f590-6eb2-45a5-a96f-dca8f1f5f883

Инстансы (3)

СОЗДАТЬ ИНСТАНС

CPU
6 из 20 шт

RAM
6144 из 10240 МБ

ОБЪЕМ ДИСКОВ
60 из 100 ГБ

IP адрес	Статус	SQL статус	Тип инстанса в облаке	Размер диска	Роль	Тип репликации	Действия
109.120.189.6	OK	SQL_OK	Basic-1-1-10 (CPU:1 RAM:1024)	10 ГБ	Ведущий		
213.219.212.93	OK	SQL_OK	Basic-1-1-10 (CPU:1 RAM:1024)	10 ГБ	Резервный	Синхронная	
89.208.221.165	OK	SQL_OK	Basic-1-1-10 (CPU:1 RAM:1024)	10 ГБ	Резервный	Асинхронная	

Создание кластера

Название *
cluster from backup

Тип инстанса в облаке *
Basic-1-1-10 (1 CPU 1024 RAM)

Версия Postgres *
PostgresProStd - 14

Размер диска (GB) *
10

Количество реплик *
0

Создание синхронной реплики

Режим создания:

Архивация wal логов

Настройки постгрес

СОХРАНИТЬ

HA

ViNA. Автоматическое переключение с мастера на реплику (стороннее средство)

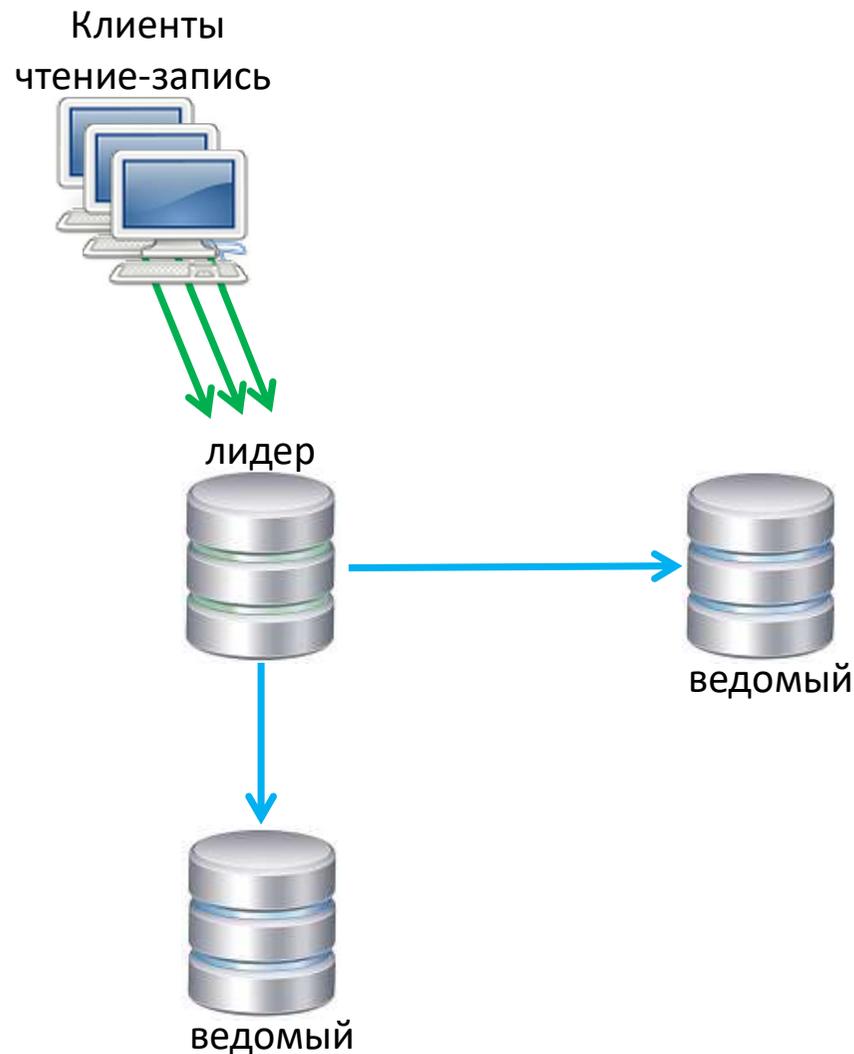
16.1

- Задача стороннего кластерного ПО при сбое мастера: обнаружить сбой, сменить роль реплики на новый мастер, но при этом не допустить работу двух узлов в режиме записи
- Примеры кластерного ПО :
Patroni, Stolon, Corosync/Pacemaker
- ПРОБЛЕМА:
 - Внешний кластер имеет **сложную архитектуру** (дополнительные узлы, сетевые каналы и т.п.)
 - Его надо **устанавливать** и конфигурировать
 - Для него тоже **требуется отказоустойчивость**
 - **Большая нагрузка** на БД может расцениваться **как отказ узла**



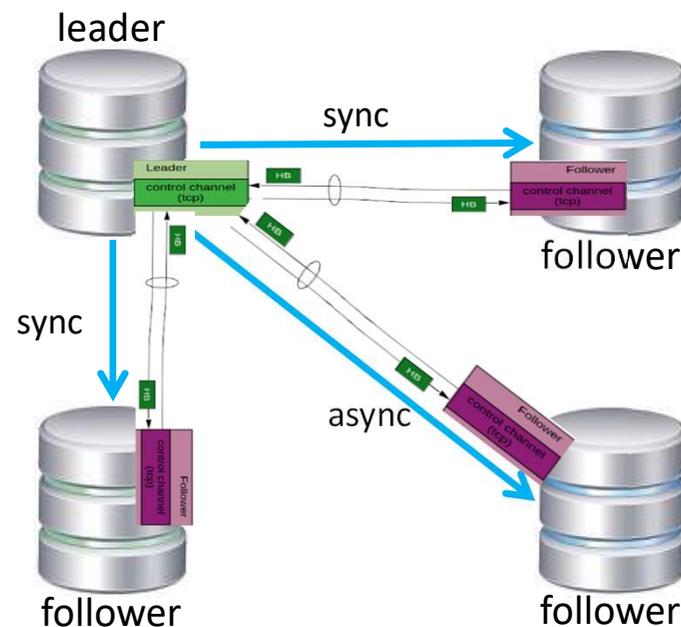
Встроенный отказоустойчивый кластер ViNA (Build-in High Availability)

- Кластерное ПО встроено в Postgres Pro
- Кластер состоит из нескольких узлов, один является лидером (leader), а другие являются фолловерами (follower) - мин 2 узла
- На разных фолловерах может использоваться как синхронная, так и асинхронная репликация.
- Клиенты могут выполнять операции чтения-записи на лидере, только чтения на фолловерах
- Непрерывный **мониторинг** состояния узлов кластера.
- Автоматическое **управление узлами** кластера (отключение проблемного фолловера, выбор нового лидера).
- **Контроль других ВНУТРЕННИХ параметров узлов** кластера (объем свободного места на диске, загруженность CPU и памяти, состояние узла). Пока для информации



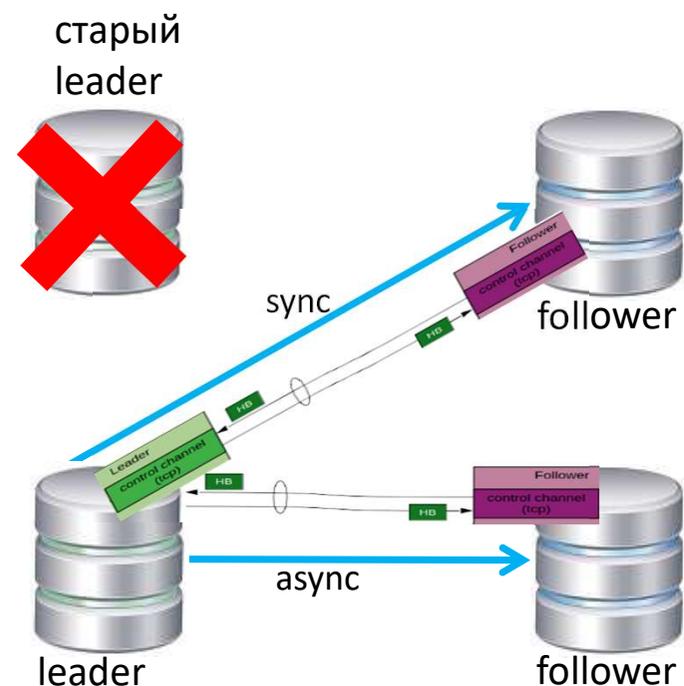
Встроенный отказоустойчивый кластер (управляющий канал)

- Намного более простая установка и конфигурирование
- Взаимодействие узлов друг с другом осуществляется с использованием управляющего канала: между любыми двумя узлами устанавливается сетевое соединение по протоколу TCP.
- При выходе из строя лидера - лидером становится флловер с максимальным WAL (т е у него мин потерь)
- При синхронной репликации – потерь нет и фоловеры голосуют
- При асинхронной репликации возможна потеря данных, а незавершенные транзакции откатятся везде



Встроенный отказоустойчивый кластер (автоматическая смена лидера)

- Автоматическая смена лидера должна происходить в аварийных ситуациях, при которых невозможна работа пользователей с БД.
- Для смены лидера связанные узлы по протоколу RAFT организуют процесс голосования. По результатам голосования происходит выбор нового лидера.
- Можно подключить и фолловеры без VINA, но управлять ими придется вручную
- На клиенте (libq, JDBC) можно указать несколько строк связи и при сбое узла будет автоматический переконнект на нового лидера



ВіНА

- Накатить журналы из архива можно вручную
- В 1 редакции НА, а не DR
- При нехватке места для WAL лидер может отключить тормозящего фолловера
- Можно задать
 - Мин количество ГОЛОСУЮЩИХ узлов
 - Мин количество РАБОТОСПОСОБНЫХ узлов
- Система должна обеспечивать доступность БД при сохранении работоспособности большей части узлов кластера. К примеру, в случае N узлов кластера, необходима работоспособность $N/2 + 1$ узлов кластера.

Pg_probackup 2.7

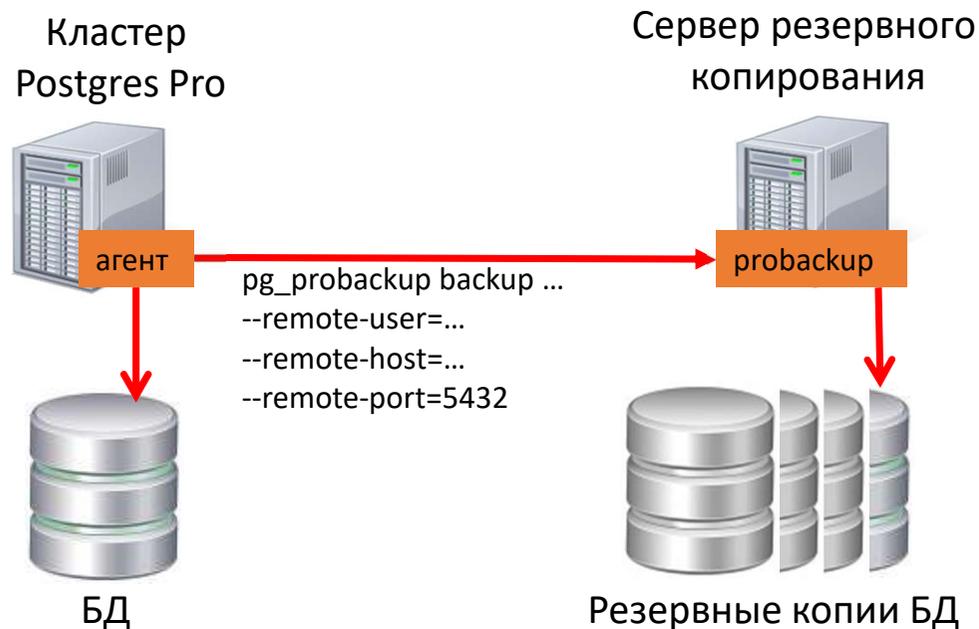
16.1

- pg_probackup обновлен до версии 2.7.0 Enterprise.
 - Улучшена производительность
 - [слияние](#) инкрементальных [резервных](#) копий сделано совместимым с системой объектного хранения S3 и выполняется теперь без создания временных локальных файлов;
 - оптимизирована работа с памятью при работе с большими базами данных: теперь для восстановления БД из нескольких тысяч таблиц и 100 инкрементальных копий требуется почти в три раза меньше памяти.

Новая версия pg_probackup – v 3

- Сегодня для каждой версии и редакции Postgres Pro собирается свой pg_probackup (т.к. структура данных меняется)
- Из каждого backup надо восстанавливаться pg_probackup той же версии в БД той же версии
- В новой версии агент для конкретной версии/редакции входит в состав ПО
- Новый формат хранения резервных копий (а не блоков)
 - Конвертер старых резервных копий из старого формата хранения в новый (например для загрузки в новую СРК) (из 2.x в 3)
- Не нужен ssh протокол (свой репликационный протокол)

Новый механизм удалённого режима работы резервного копирования по сети (без ssh)



СУБД ⇔ библиотека libpgprobackup3 ⇔ СРК или обертка

Новая версия pg_probackup

16.1

- Интеграция со сторонними СРК
- 2 редакции pg_probackup: Community и Enterprise
 - Коммерческий (Enterprise) pg_probackup только для Postgres Pro Enterprise и Шардман
 - Некоммерческий (Community) – в GitHub, бесплатно
 - Enterprise pg_probackup использует возможности Postgres Pro Enterprise
 - Сжатие CFS
 - S3
 - Интеграция с СРК (Rubackup, Киберпротект)
 - Бэкап шардированной БД
 - и т д - список уточняется
- Интеграция pg_probackup с Шардман и EM

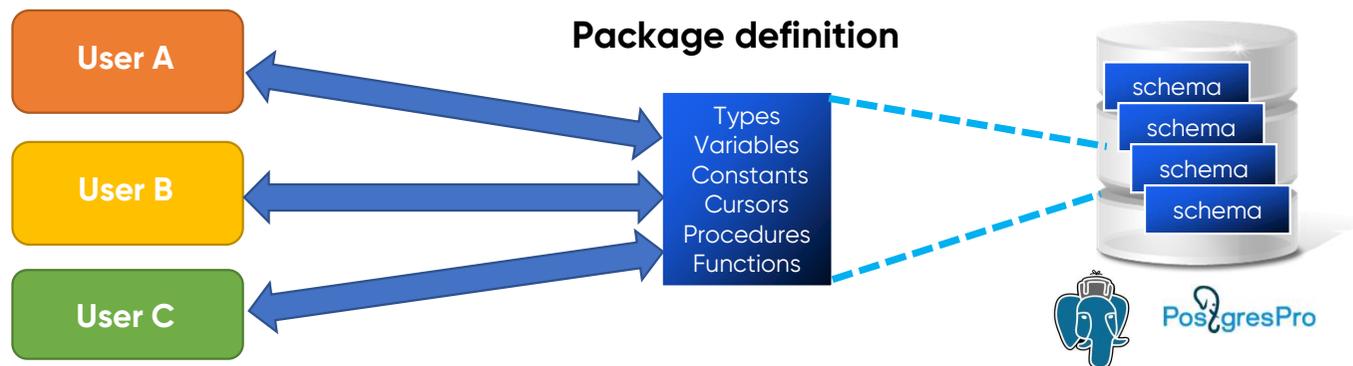
Новая версия pg_probackup

- **Быстрое** чтение карты изменений блоков PTRACK при инкрементальном резервном копировании
- Фильтрация WAL для нужной БД
 - Восстановить БД в кластере на момент сбоя можно и сейчас (восстановит потерянные файлы + накатит журналы – инкрементальное восстановление)
 - При восстановлении PITR – надо восстановить БД в новый кластер и накатить журналы, но журналы содержат инфо обо всех БД. Сейчас она накатывается, создавая новые ненужные файлы
 - Парсинг WAL (фильтрация записей WAL для нужной БД)
- Оптимизация резервного копирования CFS (сейчас Postgres Pro Enterprise при чтении CFS разжимает файл, pg_probackup передает и сжимает)
 - Новая версия будет работать без лишней распаковки и упаковки данных)

Миграция с Oracle

Поддержка пакетов PL/pgSQL в Postgres Pro 16 Enterprise (Пакеты 2.0):

- Сделано: пакеты, 4 системных пакета в Orafce/pgpro_utils, поддержка в Ora2pgpro
- В 15 версии все элементы пакета являются публичными, **В 16:** поддержка приватности элементов пакета



- Новые системные пакеты
 - UTL_HTTP,
 - UTL_MAIL
 - UTL_SMTP (функциональные аналоги соответствующих пакетов в СУБД Oracle);
- Пакет DBMS_APPLICATION_INFO для инструментирования состояния сессии и метрик выполнения длительных операций;

Поддержка работы с ассоциативными массивами (index by table - таблица pl/sql) в Postgres Pro PL/pgSQL

- В Oracle PL/SQL – 3 вида коллекций (массив, запись, **таблица PL/SQL**)
- Таблица pl/sql – это структура из 2 элементов
 - Главный ключ (целое число или **строка**)
 - Столбец (скаляр или запись)

- Ограничений на размер таблицы нет

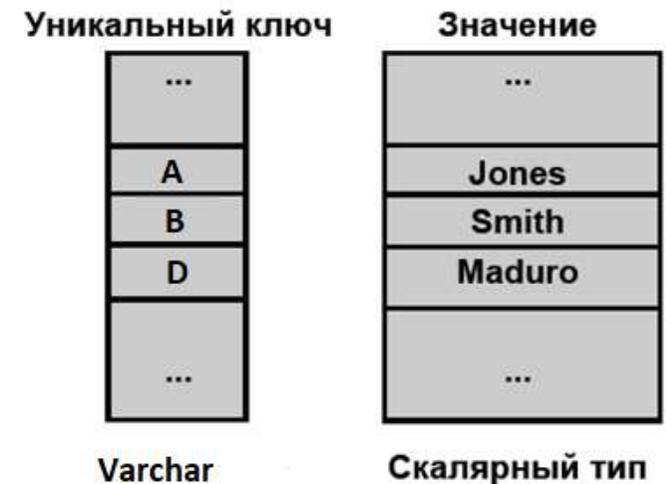
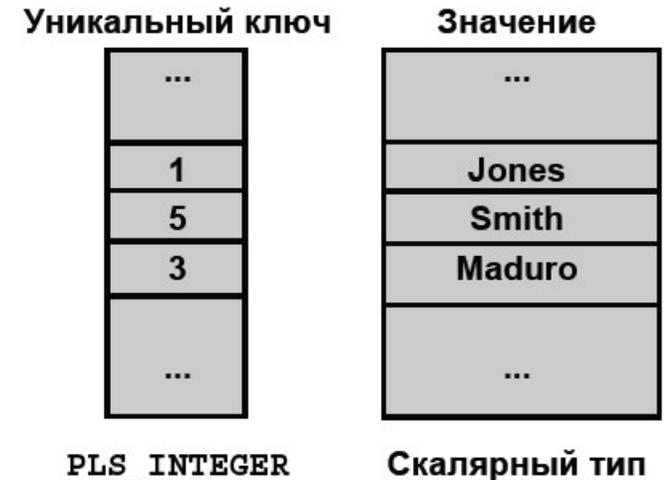
```
TYPE ename_table_type IS TABLE OF employees.last_name%TYPE  
INDEX BY PLS_INTEGER;  
ename_table ename_table_type;
```

```
Ename_table(4) := 'Иванов';
```

```
..... INDEX BY VARCHAR2(5);
```

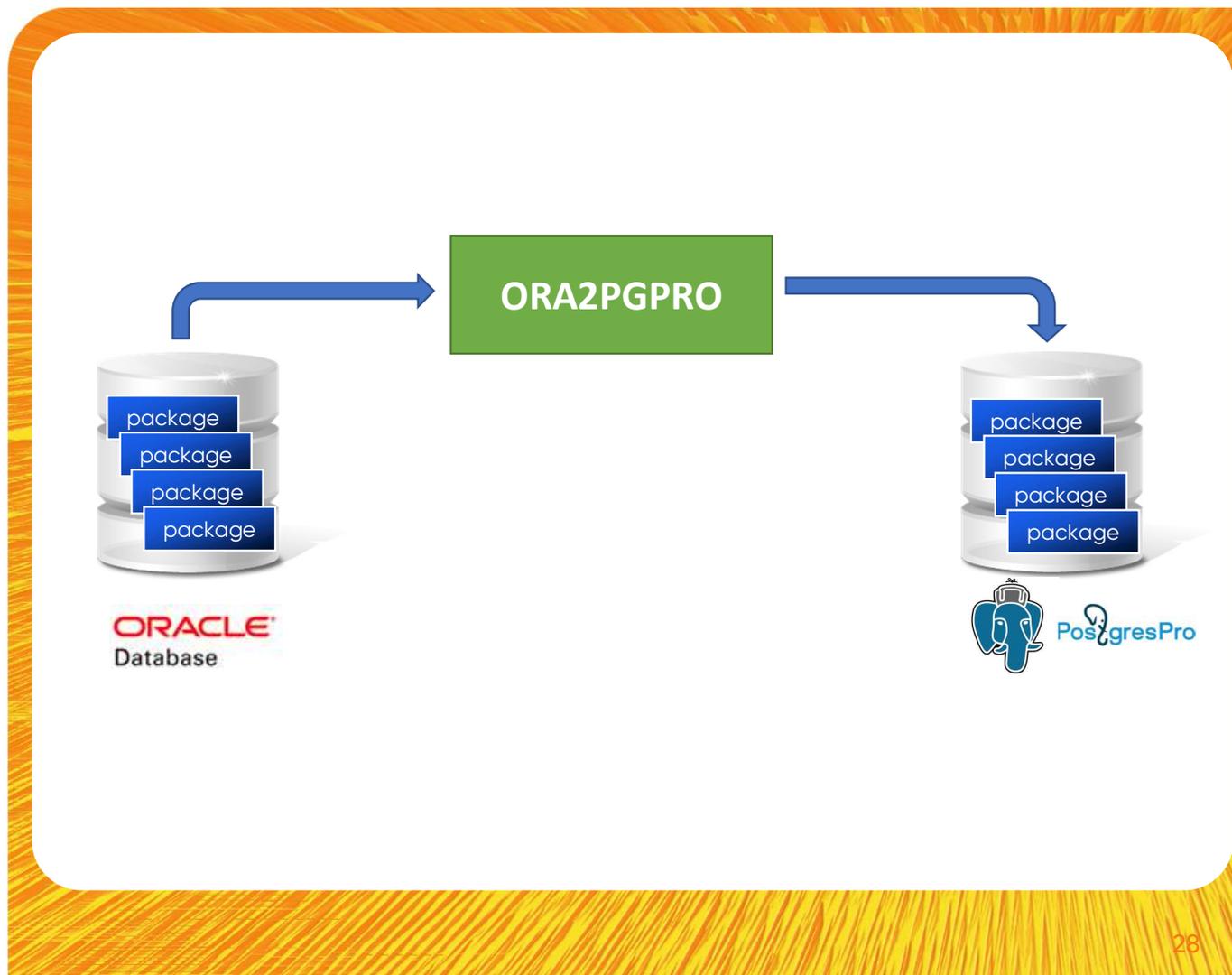
```
Ename_as_table('A') := 'Сидоров';
```

- Поддержка ассоциативных массивов/хэш коллекций (ключ-текстовый) в хранимых процедурах и функциях Postgres Pro PL/pgSQL (Таблицы PL/SQL)
- Расширение функционала расширения pg_variables – для поддержки глобальных переменных имеющих тип ассоциативного массива



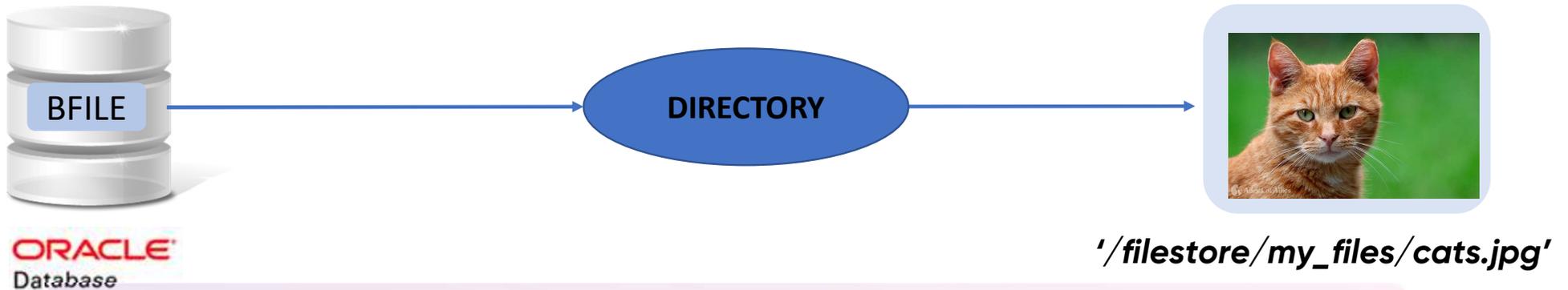
Утилита **ORA2PGPRO** – автоматическая конвертация кода пакетов

- **Автоматическая конвертация** кода работы с коллекциями Oracle PL/SQL в код Postgres Pro PL/pgSQL
- Преобразование методов массивов и pl/sql таблиц в объекты PG `mas.length` → `length(mas)`
- Автоматическая конвертация кода работы с **ассоциативными** коллекциями Oracle PL/SQL в код Postgres Pro PL/pgSQL (Таблицы PL/SQL)
- Исправление ошибок ORA2PG
- И т.д. на основе анализа недостатков конвертации



Поддержка BFile – файлов на файловой системе ОС

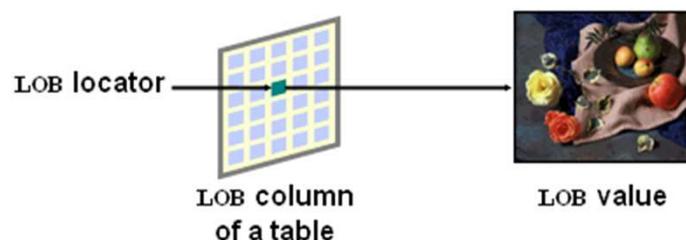
16.1



```
SQL> CREATE OR REPLACE DIRECTORY my_dir AS '/filestore/my_files/';  
  
CREATE TABLE pgpro_user.demo_bfile (Id int, image BFILE);  
  
INSERT INTO pgpro_user.demo_bfile VALUES(1, BFILENAME('MY_DIR','cats.jpg'));
```

Добавление функционала системных пакетов Oracle в Postgres Pro, работа с **BLOB** – пакет **DBMS_LOB**

- Реализация системного PL/SQL-пакета из Oracle, который отсутствует в ORAFCE:
 - DBMS_LOB
- Поддержка работы с временными большими объектами (Temporary LOB)



- Сегодня есть 2 метода работы с большими объектами TOAST (JSON, Text, Bytea) и BLOB
- Суперфайлы (замена BLOB), интерфейс в системном пакете DBMS_LOB, снятие ограничений по работе с LOB

Производительность

Resource Management Plans

- Приоритеты для пользователей и сессий при ограниченном ресурсе
- Методы выделения ресурсов:
 - CPU, чтение, запись
 - Приоритеты 1, 2, 4, 8
 - По умолчанию - 4
- Группы потребителей
 - пользователь может быть членом нескольких групп
 - только одна группа активна во время сеанса
 - группа по умолчанию назначается пользователю при соединении с БД
- Планы и приоритеты можно менять не останавливая PG
- Автоматическое назначение в группу при login
- Автоматическое/ручное переключение планов
- Сеансы работают без ограничений, если ресурсы свободны

WEEKDAY_PLAN

Resource Consumer Group	Resource Plan Directives
Order Entry	CPU = 1 Приоритет по чтению = 4
Shipping	CPU = 2 Приоритет по чтению = 4

NIGHT_PLAN

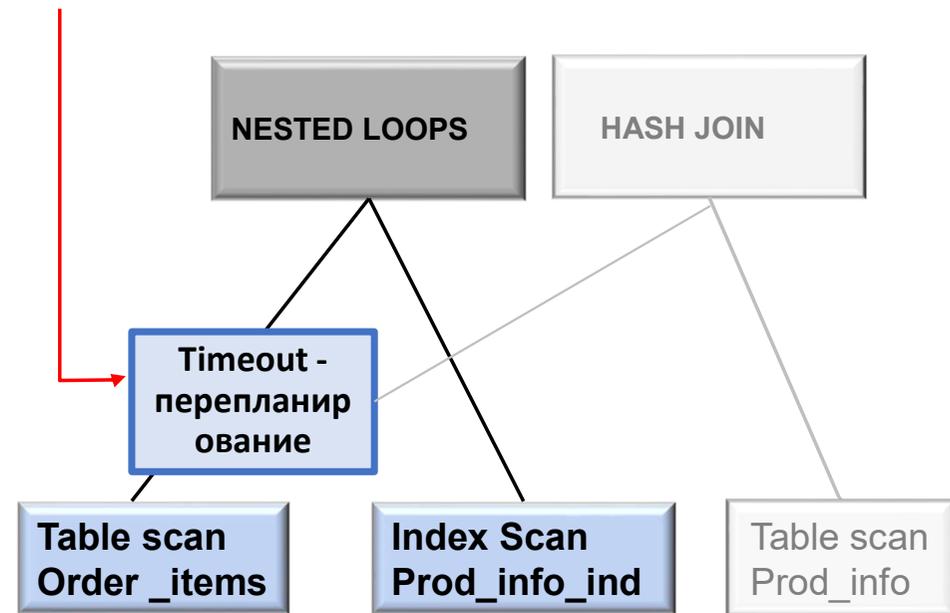
Resource Consumer Group	Resource Plan Directives
Billing	CPU = 4 Приоритет по чтению = 4
Order Entry	CPU 2 Приоритет по записи = 2
Shipping	CPU = 2 Приоритет по чтению = 2

- Настройка и использование модуля упрощены, теперь всего 3 параметра, они определяют состояние aqo, режим его работы и то, как собирается статистика :
 - **aqo.enable** – включить/выключить aqo на уровне экземпляра или сессии
 - **aqo.mode** - обучаться и использовать знания/только использовать знания
 - **aqo.advanced** - обучаться целиком по запросам или отдельным узлам плана
- По умолчанию устанавливается режим, в котором данные машинного обучения собираются для отдельных узлов плана и собранные знания будут использоваться для **всех запросов**, в планах которых присутствуют эти узлы
- Модуль aqo теперь может работать независимо с несколькими базами данных
- Чтобы мгновенно отключить/включить aqo на уровне базы данных, теперь достаточно выполнить команду DROP/CREATE EXTENSION
- В 2 раза уменьшено потребление памяти и накладные расходы при планировании

Адаптивный Executor (Replanner)

- Если оптимизатор запросов «ошибся» при выборе плана – он может исправиться прямо во время выполнения запроса
- Если запрос долго выполняется и статистика неверна, то он прерывается и передается оптимизатору для генерации нового плана с учетом накопленной статистики, далее запрос уходит на повторное выполнение
- В новом плане могут поменяться методы доступа к данным, типы и порядок соединений и т.д.

Например, при повторном планировании оптимизатор решил поменять метод соединения с nested loops на hash join



По умолчанию выбран **nested loops join**

- Добавлены системные представления в словарь (без extension)

- `pg_stats_vacuum_tables`,
- `pg_stats_vacuum_indexes`
- `pg_stats_vacuum_database`,

Они показывают статистики очистки таблиц, индексов и баз данных
Ранее эти статистики были доступны только в `pgpro_stats_`.

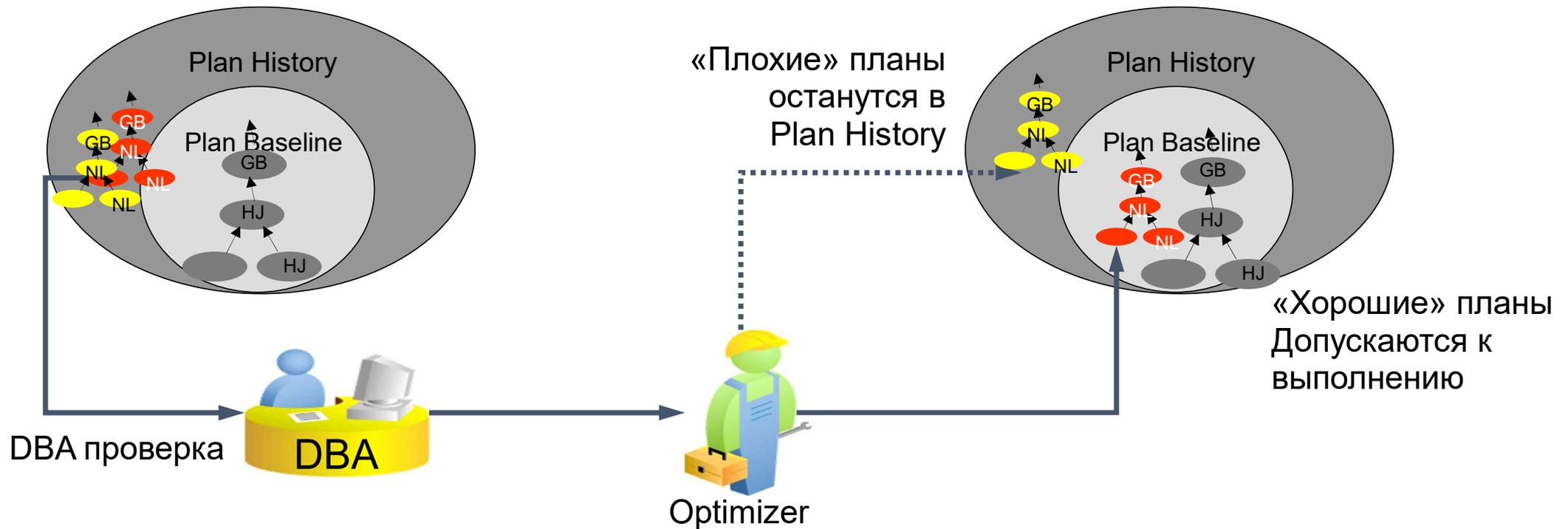
- Расширение `pg_proaudit 2.0`.
 - Объединение пользователей в группы
 - В СУБД добавлены фиксированные группы контролируемых аудитом действий (`ALL_DDL`, `ALL_DML`, `ALL_PROC`, `ALL_ROLE`)
 - Назначение группы контролируемых действий группе пользователей

Стабилизация планов запросов – **baselines**

- План – способ выполнения запроса
- Для одного запроса может быть несколько планов, в зависимости от условий
- Какие-то планы могут быть не оптимальными – производительность ухудшится
- Baselines – механизм, в котором новые планы сохраняются в список, но не выполняются, а ждут проверки со стороны DBA

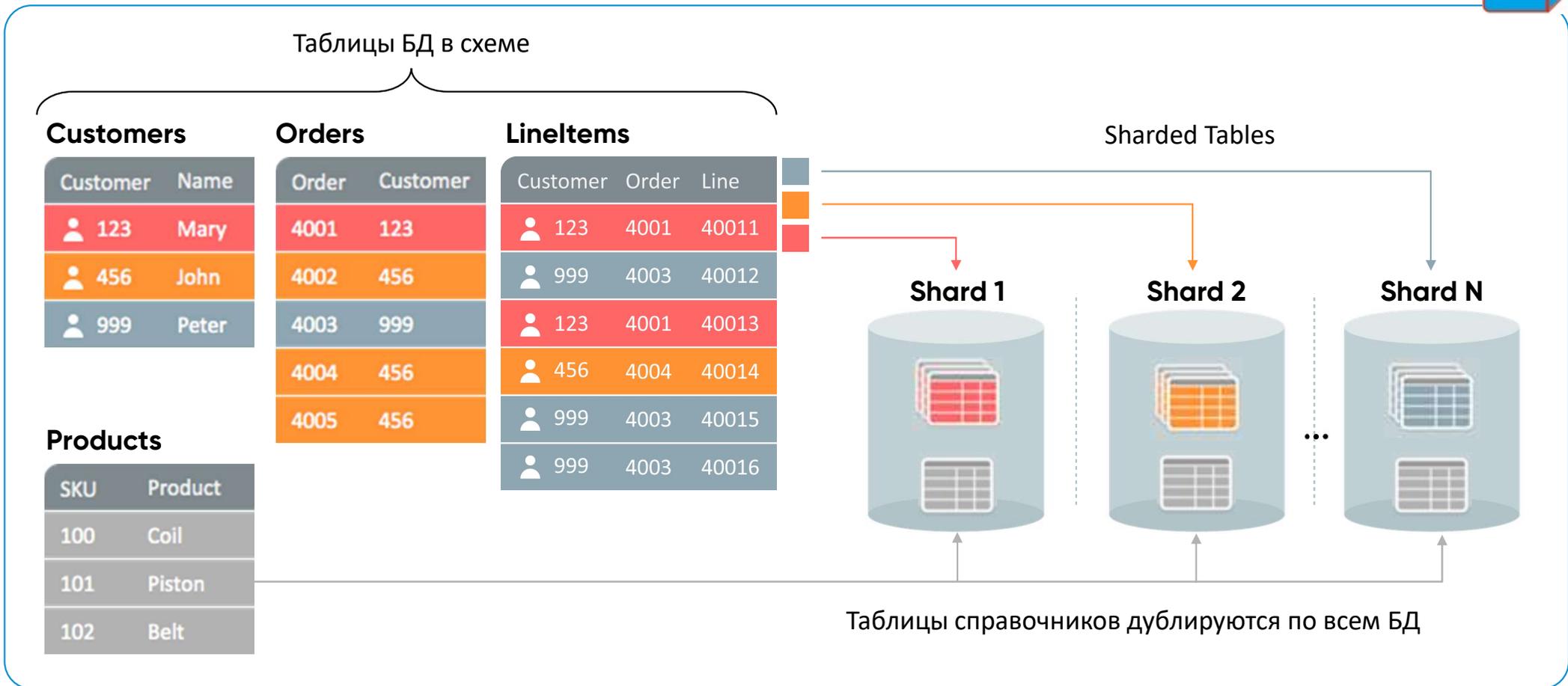
Стабилизация планов запросов

- Схематично:



Шардирование

Шардирование по хэш функции снимает ограничения на размер БД



Шардман

- Параллельная обработка, перенос подзапросов на другие шарды и сбор результата
- Согласованный бэкап и восстановление шардов
- Загрузчик шардов из нешардированной БД
- Шардинг для OLTP т к нет координатора (накладные расходы, надежность)
- Каждый шард имеет реплику
- Можно подсоединиться к любому шарду
- Полноценная документация, мониторинг через EM
- Полная поддержка распределенных транзакций (у аналитических систем возможно чтение незафиксированных изменений – Citus)

Шардинг vs шардинг

- Шардман - OLTP
- CИTUS – аналитика
- Шардман – согласованность шардов при транзакции
- Любой узел – координатор, нет единой точки

Безопасность

Новые требования ФСТЭК

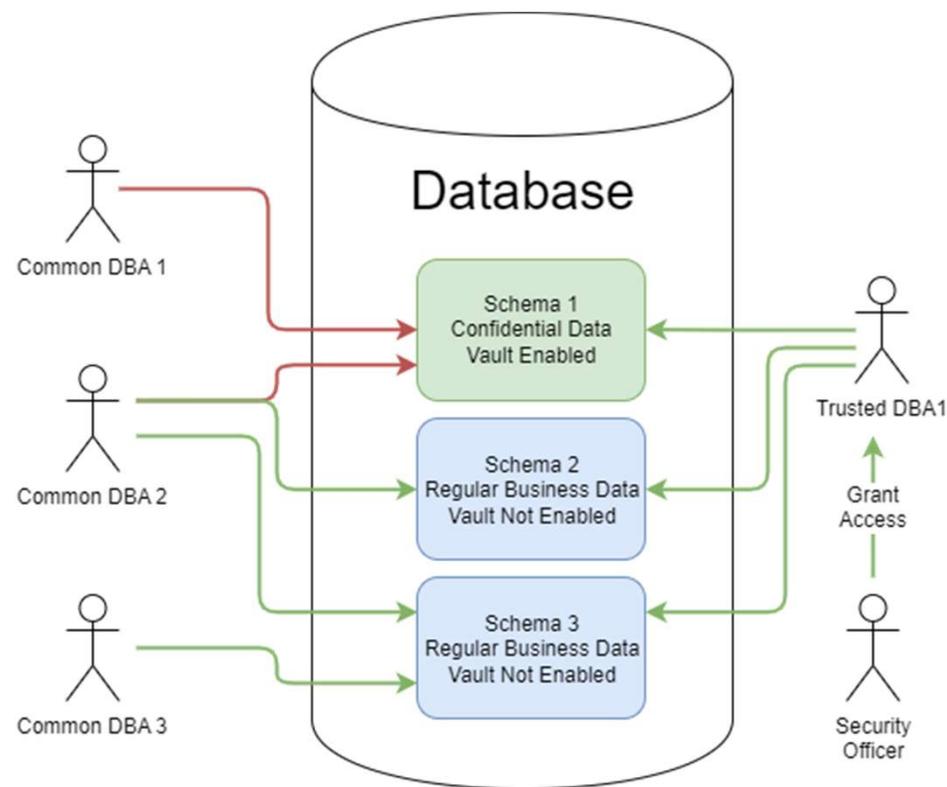
16.1

- **Права администраторов ограничены.** Администратор не может в одиночку незаметно выполнить действия с конфиденциальной информацией
 - Не может прочитать её незаметно
 - Не может изменить её незаметно
- **Администраторов теперь два**
 - Администратор СУБД – настраивает сам сервер
 - Администратор БД – настраивает конкретную БД на сервере
- **Пользователям запрещены** любые изменения кода информационных систем
 - Пользователи не могут создавать и изменять код хранимых процедур и функций
- **Чёткие требования к логированию**
 - Любые действия с пользователями
 - Все изменения прав пользователей
 - Изменение конфигурации СУБД
 - Изменение хранимок
 - Любой DDL

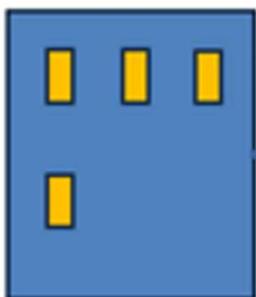
Админ без доступа к данным

16.1

- защита от доступа привилегированных пользователей к бизнес-данным
- ограничение полномочий администраторов (SUPERUSER) по созданию объектов и модификации данных (DDL, DML, DCL)
- Защищенная схема (зона)
- полномочия администратора (SUPERUSER) разделены между ролями **менеджер прав доступа** зоны и **владельца данных** зоны
- механизм ограничения полномочий по созданию и назначению ролей при наличии привилегий по назначению ролей (защита от опосредованного повышения привилегий)

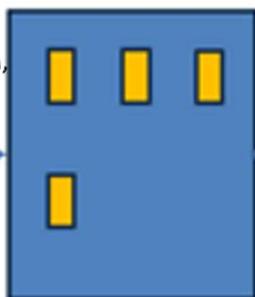


схема



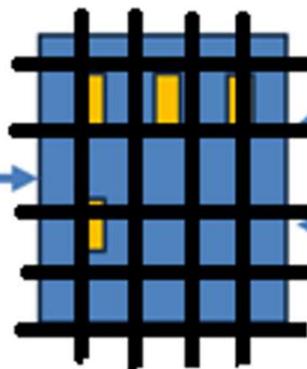
Суперпользователь под контролем, Технологический режим

схема



Отключить супер-Пользователя и технологический режим

Зона повышенной безопасности



Администратор



Пользователь



Владелец: DB_ADMIN



Владелец: Менеджер зоны



Владелец: Менеджер зоны



Менеджер прав доступа: Нет



Менеджер прав доступа: Менеджер прав доступа

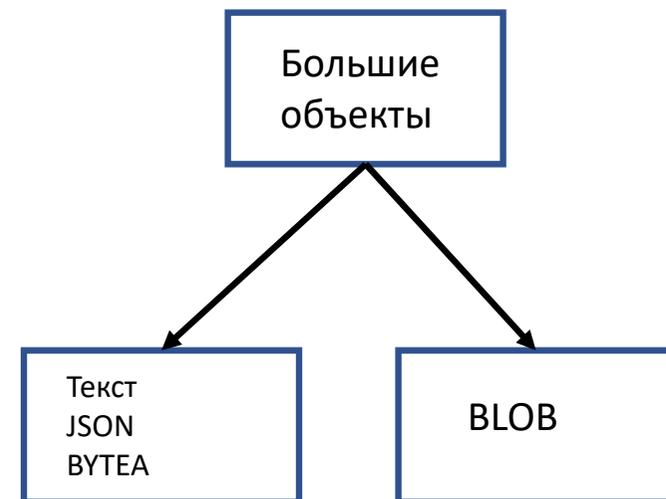


Менеджер прав доступа: Менеджер прав доступа

Разработка

Работа с большими объектами (версия 15)

- Медленно работает
- TOAST (The Oversized-Attribute Storage Technique)
 - используем SQL
 - Text, Json, Bytea, Video, Audio, рисунки
 - Ограничения:
 - размер значения (1Гб до сжатия),
 - суммарный размер больших полей таблицы (32Тб) – т к 1 TOAST на все такие колонки таблицы
 - количество значений больших полей в таблице (2^{32}), при коротких значениях это ограничит до 8 Тб (а не 32 Тб)
 - UPDATE сводится к DELETE+INSERT и место освободится только после Vacuum
 - Непредсказуемая производительность
- Один механизм хранения и обработки для всех типов данных неоптимален
- BLOB
 - Ограничения:
 - Размер 4 Тб
 - Всего 32 Тб на БД (т е 8 BLOB)
 - Работа через API



Pluggable TOAST (для text, json, Bytea)

16.1

- Разные механизмы и способы хранения и обработки для разных типов данных (Тостер)
- Можно писать свой тостер
- Скорость работы выше
- Снимем ограничение 32 Tb на VCE колонки таблицы

- Тостеры:
 - bytea appendable Toaster: производительность добавления данных в bytea зависит только от объёма вставляемых данных (для Append и Update типа date=date ||)
 - jsonb Toaster: оптимальное хранение и высокая производительность работы с данными json
 - Оптимизация обновления частей (JSONB)
 - Частичная декомпрессия при чтении TOAST фрагмента (как только дошли до нужного элемента – stop)

Суперфайлы (замена BLOB)

- Хранение неограниченного количества значений (4 млрд) размером до 4 Тб
- Ограничение в 4 млрд записей на таблицу (а не на всю БД)
- Объекты хранятся в своей инфраструктуре (системе таблиц, скрытой от пользователя)
- Компрессия и дедупликация как на уровне объекта целиком, так и на уровне его блоков
- Ускорение чтения и добавления в конец
- Функциональный интерфейс, близкий к возможностям пакета Oracle `dbms_lob`

Из PostgreSQL 16

Полезные механизмы унаследованы из PostgreSQL 16

- Логическая репликация:
 - Репликация из Standby (read only, разгрузка мастера)
 - Двухнаправленная репликация одной таблицы (могут быть конфликты)
 - Роль `pg_create_subscription`
- Использование SIMD команд при поиске в Json, ASCII
- В таблице `pg_stat_all_tables` – время последнего использования таблицы/индекса
- В `pg_hba.conf` и `pg_ident.conf` вместо имен пользователей и БД можно использовать регулярные выражения (т е работать с группой имен)
- Сору ускорили иногда в 3 раза
- В `psql` можно писать команды с параметрами
 - `Select $1, $2\bind 5 10\g`

Миграция на Postgres Pro Enterprise 16

- Вы можете перейти на Postgres Pro Enterprise 16 с той же или предыдущей версии
 - PostgreSQL (которая поддерживается выбранным способом обновления)
 - Postgres Pro Standard/Postgres Pro Standard Certified
 - предыдущей версии Postgres Pro Enterprise/Postgres Pro Enterprise
- Способы обновления кластера базы данных описаны в [Разделе 18.6](#) документации.
- Обратный переход не поддерживается.
- Для перехода с PostgreSQL, Postgres Pro Standard или выпуска Postgres Pro Enterprise, основанного на предыдущей основной версии PostgreSQL, сначала установите его последний корректирующий выпуск, а затем выполните выгрузку/восстановление данных, применив [pg_dumpall](#), или воспользуйтесь [pg_upgrade](#).
- Если вы решите использовать `pg_upgrade`, важно инициализировать новый кластер баз данных с совместимыми параметрами. В частности, обратите внимание на характеристику [контрольных сумм](#) в кластере, который вы будете обновлять. Если `pg_upgrade` создаст какие-либо скрипты SQL в текущем каталоге, выполните их для завершения обновления.

PostgresPro

Q & A

