



# Опыт внедрения SDL в компании Postgres Professional

Попов Валерий Викторович

Руководитель отдела ИБ

## 8 лет

на рынке с 2015 г.

## >20 лет

опыта в разработке PostgreSQL

## >1,7 млрд руб.

объем инвестиций в развитие СУБД Postgres Pro

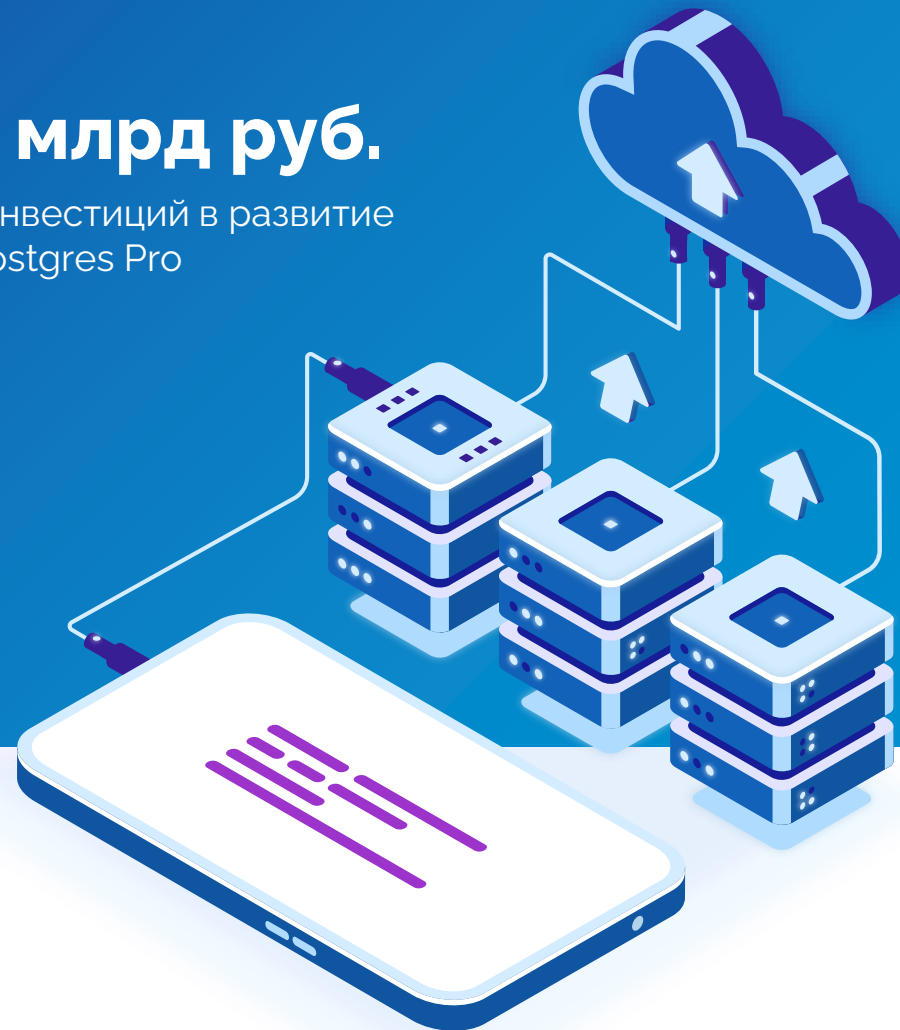
## >180 чел.

штат компании

включая ведущих разработчиков и коммитеров

## >100 патчей

направляют сотрудники компании ежегодно в международное сообщество PostgreSQL



# ПРОДВИЖЕНИЕ POSTGRES

Организация крупнейшей в мире ежегодной конференции по PostgreSQL – PGConf.Russia

Книги, учебные пособия, HABR – 42K подписчиков, больше 150 статей

Поддержка документации PostgreSQL на русском языке

Сотрудничество с ВУЗами в области преподавания дисциплин по технологиям баз данных



# СУБД POSTGRES PRO

**СУБД Postgres Pro – первый в России коммерческий продукт на основе PostgreSQL.**

**Входит в Единый реестр отечественных программ и баз данных Минкомсвязи**

## Standard

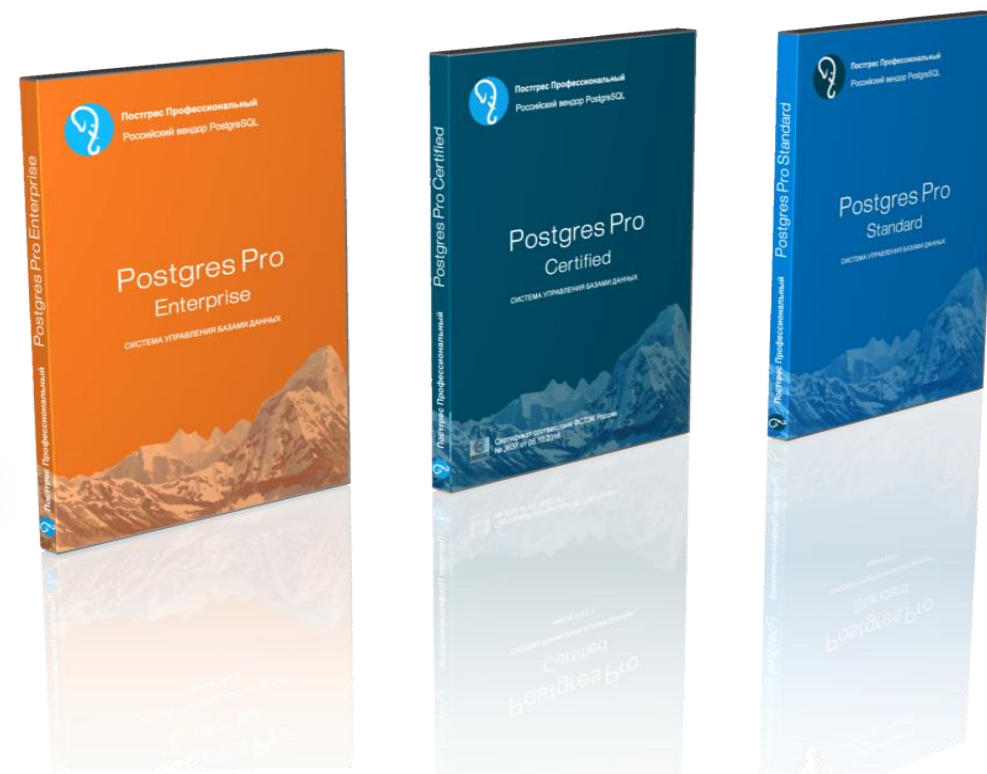
Современная СУБД, включает все новые функции PostgreSQL и полезные доработки от компании

## Enterprise

Наиболее полнофункциональная СУБД с высокой производительностью и масштабируемостью

## Certified

Сертифицированные ФСТЭК версии Standard и Enterprise



# ОТЛИЧИЯ POSTGRES PRO (ENT) ОТ POSTGRES SQL

## ПРОИЗВОДИТЕЛЬНОСТЬ

**JSONB:**  
Запросы к полям jsonb

**CFS:**  
Сжатие данных

**SR\_PLAN:**  
Сохранение плана запроса

**Автономные транзакции**

**Инкрементальный бэкап**

**Поддержка MS SQL**

## НАДЕЖНОСТЬ

**64-XID**

**Scheduler**

**До 150 ТБ:**  
Размер базы данных

**До 10000**  
пользователей

# НАШИ SDL-ПРОЦЕССЫ

**1**

Все изменения в коде проверяются на rpgfarm

**3**

Регрессионное тестирование > 400 тестов

**5**

Динамическое тестирование – набор средств

**7**

Ежеквартальные выпуски обновлений

**2**

Статический анализ с помощью Svnace

**4**

Safe-compiler

**6**

Обучение и семинары

# РЕГРЕССИОННОЕ ТЕСТИРОВАНИЕ

## ПРИ ЛЮБОМ ИЗМЕНЕНИИ КОДА

Каждая задача разрабатывается в отдельной ветке в gitlab, затем вливается в DEV, STABLE

```
=====  
All 207 tests passed.  
=====
```

## ПРИ РЕЛИЗНОМ ТЕСТИРОВАНИИ

Цель – добиться полного выполнения всех тестов

PostgreSQL BuildFarm

Home Status Failures Members Register Typedefs Branches SSH Client

### PostgreSQL BuildFarm Status

Shown here is the latest status of each farm member for each branch it has reported on in the last 30 days.

Use the farm member link for history of that member on the relevant branch.

Legend: = cassari = debug = disable-integer-datetimes = gssapi = krb5 = nls = openssl  
 = pam = perl = python = tcl = thread-safety = vpath = xml

Alias	System	Status	Flags
altlinux-9	AltLinux 9 gcc 8.3.1 x86_64	00:21:53 ago OK [d56debe] Config	
rhel-9-arm	RHEL 9 gcc 11.2.1 aarch64	00:37:14 ago OK [d56debe] Config	
altlinux-spt-82	ALT SP 8.2 gcc 8.4.1 x86_64	00:38:53 ago OK [d56debe] Config	
rosa-sx-7	ROSA Enterprise Linux Cobalt 7.3 gcc 4.8.5 x86_64	00:41:53 ago OK [d56debe] Config	
astra-novorossiysk	Astra Linux SE (Novorossiysk) 4.7.0 gcc 8.3.0 arm64	00:45:02 ago OK [d56debe] Config	
buster-arm	Debian 10 gcc 8.3.0 arm64	00:46:24 ago OK [d56debe] Config	
redos-7.3	RedOs 7.3 gcc 8.3.1 x86_64	00:48:59 ago OK [d56debe] Config	
bookworm	Debian 12 gcc 12.2.0 amd64	00:52:04 ago OK [d56debe] Config	
rhel-9	RHEL 9 gcc 11.2.1-9 x86_64	00:53:59 ago OK [d56debe] Config	
sles15	SUSE Linux ES 15 gcc 7.3.1 x86_64	00:54:38 ago OK [d56debe] Config	
bullseye32	Debian 11 gcc 10.2.1 i386	00:55:03 ago OK [d56debe] Config	
centos8	Centos 8 gcc 8.2.1 x86_64	00:55:35 ago OK [d56debe] Config	
buster32	Debian 10 gcc 8.3.0 i386	00:55:36 ago OK [d56debe] Config	
jammy	Ubuntu 22.04 gcc 11.2.0 amd64	00:59:41 ago OK [d56debe] Config	
bookworm-arm	Debian 12 gcc 12.2.0 arm64	01:04:05 ago OK [d56debe] Config	
astra-smolensk-1.7	Astra Linux SE (Smolensk) 1.7.0 gcc 8.3.0 x86_64	01:05:20 ago OK [d56debe] Config	
jammy-arm	Ubuntu 22.04 gcc 11.2.0 arm64	01:13:56 ago OK [d56debe] Config	
rhel8	RHEL 8 gcc 8.2.1 x86_64	01:14:30 ago OK [d56debe] Config	
centos7	CentOS 7 gcc 4.8.5 x86_64	01:19:25 ago OK [d56debe] Config	

# СТАТИЧЕСКИЙ АНАЛИЗ КОДА ИНСТРУМЕНТОМ SVACE

CI/CD GITLAB

РЕГУЛЯРНАЯ ПРОЦЕДУРА

ЗАДАЧИ НА ИСПРАВЛЕНИЕ В JIRA:

НЕСКОЛЬКО ДЕСЯТКОВ В КВАРТАЛ

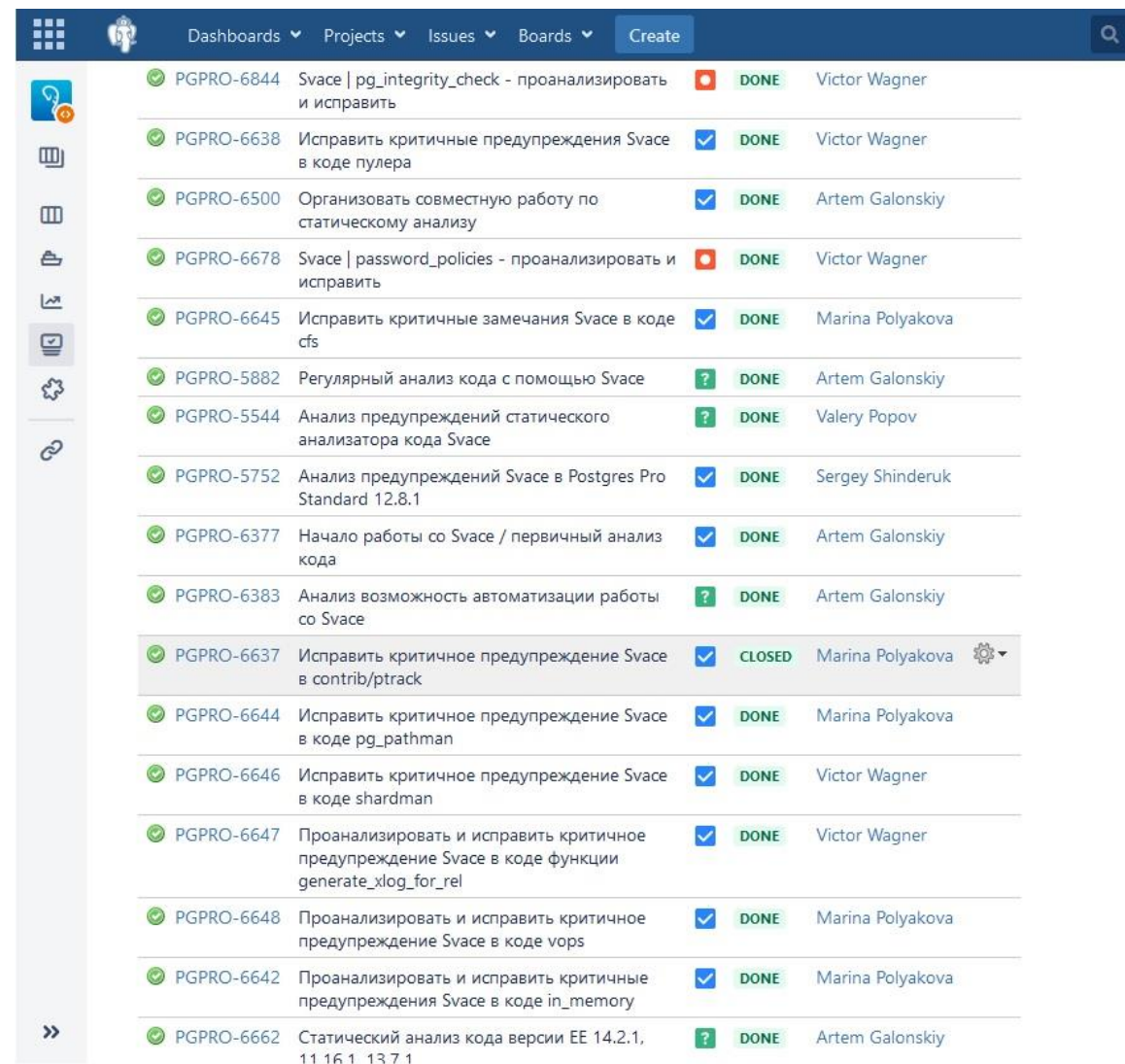
ТРЕБУЕТСЯ ВЫСОЧАЙШАЯ

КВАЛИФИКАЦИЯ ДЛЯ РАЗМЕТКИ

НЕОБХОДИМО РАЗДЕЛЯТЬ УСИЛИЯ

СООБЩЕСТВА ДЛЯ РАЗМЕТКИ

POSTGRESQL



ID	Task Description	Status	Assignee
PGPRO-6844	Svace   pg_integrity_check - проанализировать и исправить	DONE	Victor Wagner
PGPRO-6638	Исправить критичные предупреждения Svace в коде пулера	DONE	Victor Wagner
PGPRO-6500	Организовать совместную работу по статическому анализу	DONE	Artem Galonskiy
PGPRO-6678	Svace   password_policies - проанализировать и исправить	DONE	Victor Wagner
PGPRO-6645	Исправить критичные замечания Svace в коде cfs	DONE	Marina Polyakova
PGPRO-5882	Регулярный анализ кода с помощью Svace	DONE	Artem Galonskiy
PGPRO-5544	Анализ предупреждений статического анализатора кода Svace	DONE	Valery Popov
PGPRO-5752	Анализ предупреждений Svace в Postgres Pro Standard 12.8.1	DONE	Sergey Shinderuk
PGPRO-6377	Начало работы со Svace / первичный анализ кода	DONE	Artem Galonskiy
PGPRO-6383	Анализ возможность автоматизации работы со Svace	DONE	Artem Galonskiy
PGPRO-6637	Исправить критичное предупреждение Svace в contrib/ptrack	CLOSED	Marina Polyakova
PGPRO-6644	Исправить критичное предупреждение Svace в коде pg_pathman	DONE	Marina Polyakova
PGPRO-6646	Исправить критичное предупреждение Svace в коде shardman	DONE	Victor Wagner
PGPRO-6647	Проанализировать и исправить критичное предупреждение Svace в коде функции generate_xlog_for_rel	DONE	Victor Wagner
PGPRO-6648	Проанализировать и исправить критичное предупреждение Svace в коде vops	DONE	Marina Polyakova
PGPRO-6642	Проанализировать и исправить критичные предупреждения Svace в коде in_memory	DONE	Marina Polyakova
PGPRO-6662	Статический анализ кода версии EE 14.2.1, 11.16.1, 13.7.1	DONE	Artem Galonskiy



# ДИНАМИЧЕСКОЕ ТЕСТИРОВАНИЕ

- **PG\_REGRESS\_FUZZING**
- **ГЕНЕРАЦИОННЫЙ ФАЗЗИНГ Sqlancer и Squirrel**
- **ФАЗЗИНГ СЕТЕВОГО ПРОТОКОЛА (libpq)**
- **ФАЗЗИНГ input-функций типов данных (jsonb, int, line, varchar,...)**
- **ФАЗЗИНГ ОПЕРАЦИЙ НАД ТИПАМИ (Structure Aware Fuzzing)**
- **FUTAG: автоматическая генерация целей для libpq**
- **CRUSHER, AFL++**

# PG\_REGRESS\_FUZZ

Основная идея: **мутирование** регрессионных тестов

Меняем строки местами, удаляем строки, переставляем и удаляем блоки текста.

Результаты: **CVE-2019-10164, CVE-2022-2625**

Десятки багов, принятых в сообщество

```
$ git log | grep Lakhin | wc -l  
142
```

# ГЕНЕРАЦИОННЫЙ ФАЗЗИНГ

ПОЗВОЛЯЕТ ГЕНЕРИРОВАТЬ СЛУЧАЙНЫЕ SQL ЗАПРОСЫ НА ОСНОВЕ ГРАММАТИК

## SQLancer (Synthesized Query Lancer)



## SQuirreL SQL Client

Нашли несколько ошибок, например,

```
Select '' similar to '\5';
```

В регулярном выражении стоит не обычная 5, а U+FF15 (Fullwidth Digit Five).



# ФАЗЗИНГ СЕТЕВОГО ПРОТОКОЛА

## Эмулирование сетевого взаимодействия

Переопределение системных функций сетевого стека. Postgres работает в `single mode`, но вместо сети общается с фаззером.

## FUTAG

Автоматически выделены и были протестированы функции `libpq` для всех версий:

*PQescapeBytea, PQfinish PQinitSSL pg\_get\_encoding\_from\_locale pg\_ascii\_toupper  
pg\_clean\_ascii*

# ФАЗЗИНГ ТИПОВ И ОПЕРАЦИЙ (> 120 ЦЕЛЕЙ)

## CBSES – система автоматического запуска исследований

- ПОДГОТОВКА ИСХОДНОГО КОДА
- СБОРКА С САНИТАЙЗЕРАМИ
- ФАЗЗИНГ
  - В режиме статической инструментации с санитайзерами ASAN и UBSAN
  - В режиме динамического символьного выполнения DSE с опцией –optimistic
  - В режиме динамического символьного выполнения с предикатами безопасности
  - Построение покрытия для найденных образцов

## Structure Aware Fuzzing

- Библиотека Libblobstamper
  - Генерирует валидные входные данные сложной структуры для операндов, чтобы усилия фаззинга направлялись на саму операцию

# ГДЕ ВЗЯТЬ ЛЮДЕЙ?


**СВОИ КАДРЫ МЫ ВЫРАЩИВАЕМ САМИ**

**ХОРОШЕЕ ПОНИМАНИЕ РАБОТЫ С ФАЗЗЕРАМИ**

**ДЛЯ СТАТИЧЕСКОГО АНАЛИЗА НУЖЕН УРОВЕНЬ SENIOR**

**ГЛУБОКОЕ ПОНИМАНИЕ РАБОТЫ СУБД ИЗНУТРИ**

# ЧТО МЫ ЖДЕМ ОТ РЕГУЛЯТОРА?



**СЕРТИФИКАЦИЯ ПРОИЗВОДСТВА ДЛЯ ОБЛЕГЧЕНИЯ  
СЕРТИФИКАЦИИ ПРОДУКТА И УМЕНЬШЕНИЯ ОБЪЕМА  
ПРОТОКОЛОВ**



**ТРЕБОВАНИЯ ПО БЕЗОПАСНОСТИ К СУБД**

Протестировать  
СУБД Postgres Pro:



117036, Москва, ул. Дмитрия Ульянова, 7А



8 (495) 150-06-91



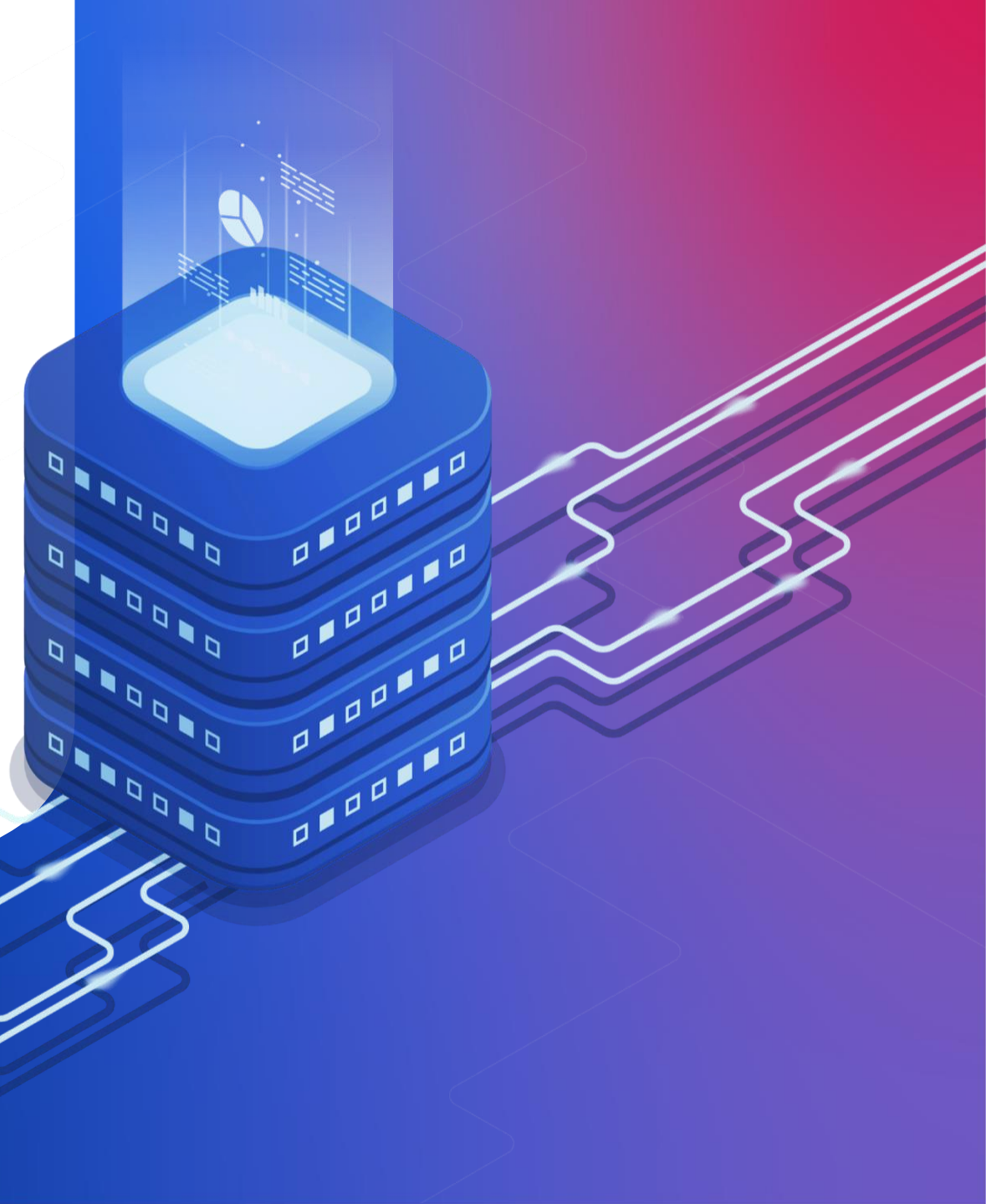
[sales@postgrespro.ru](mailto:sales@postgrespro.ru)

[postgrespro.ru](http://postgrespro.ru)



PosgresPro

**Спасибо  
за внимание!**



PosgresPro

# Q&A

