



PROFESSIONAL  
**Posgres**

**WAITING FROM  
POSTGRESQL**

**10**

**Дорошкевич Антон**  
Инфософт, Руководитель ИТ, Новосибирск



**PGConf.Russia**



# Support PG PRO — тикетная система

+ Новая заявка

🔔 Мои заявки

📁 Заявки компании

👤 Компания

📄 Мои данные

📄 Отчеты



⊕ Настройка фильтрации

Все заявки



№ заявки

Тема:

Статус: ▼



000022 В меню

Открыта



000021 Не работает принтер

Закрота



000020 проверка объектов

Закрота



000016 Не работает

Открыта



000015 Вернуть

Открыта



000009 Обновить

Закрота



000007 Прошу провести демонстрацию

Закрота



000002 Заменить

Закрота

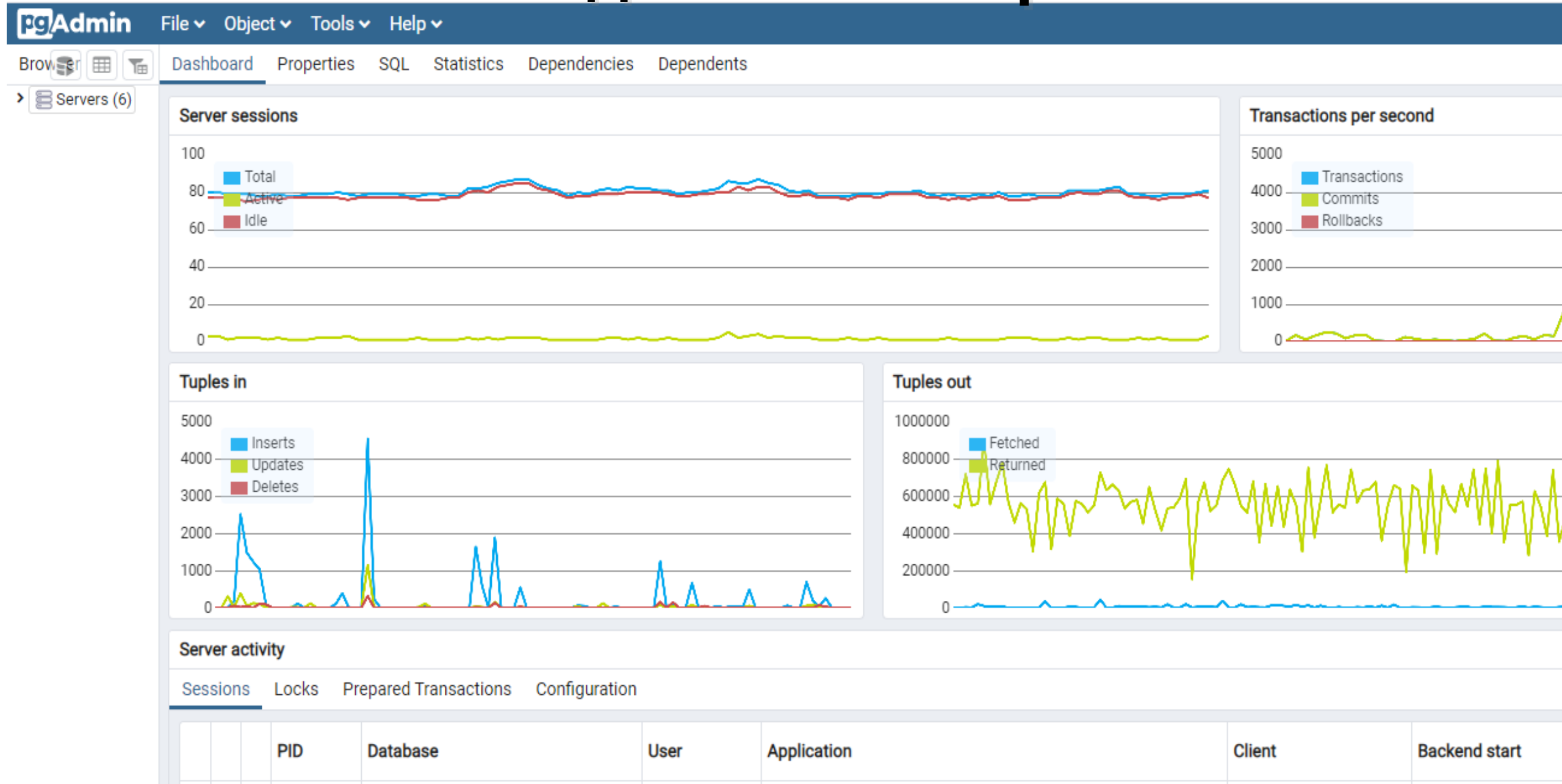


000001 Установить

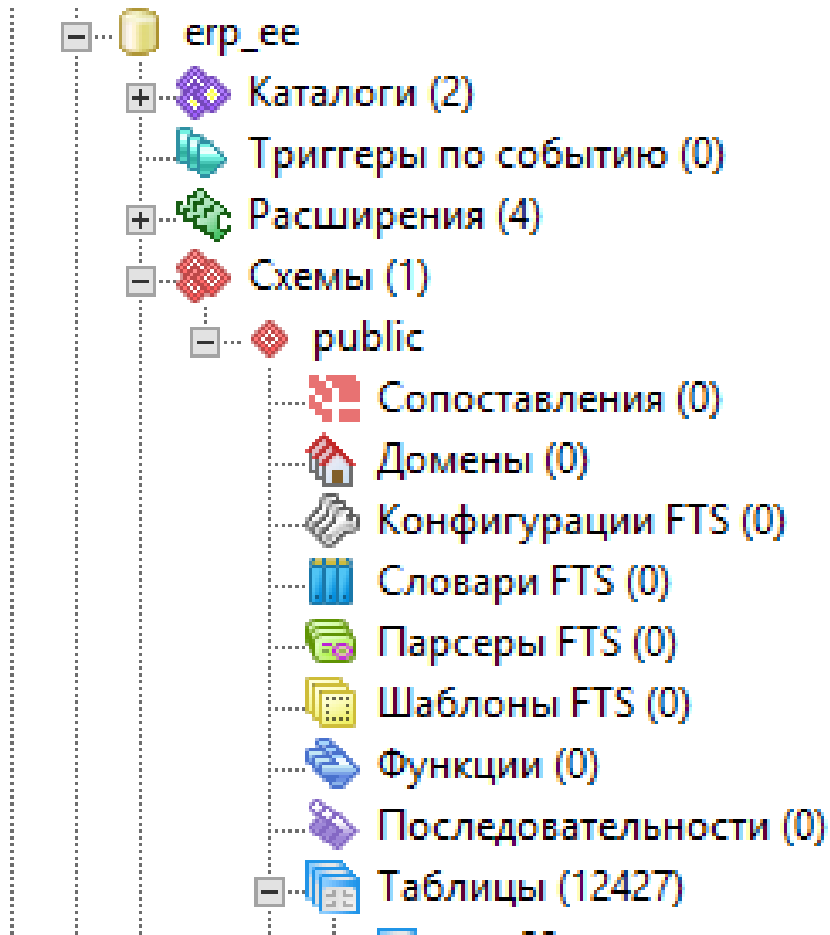
Закрота



# PGAdmin — десктопное приложение



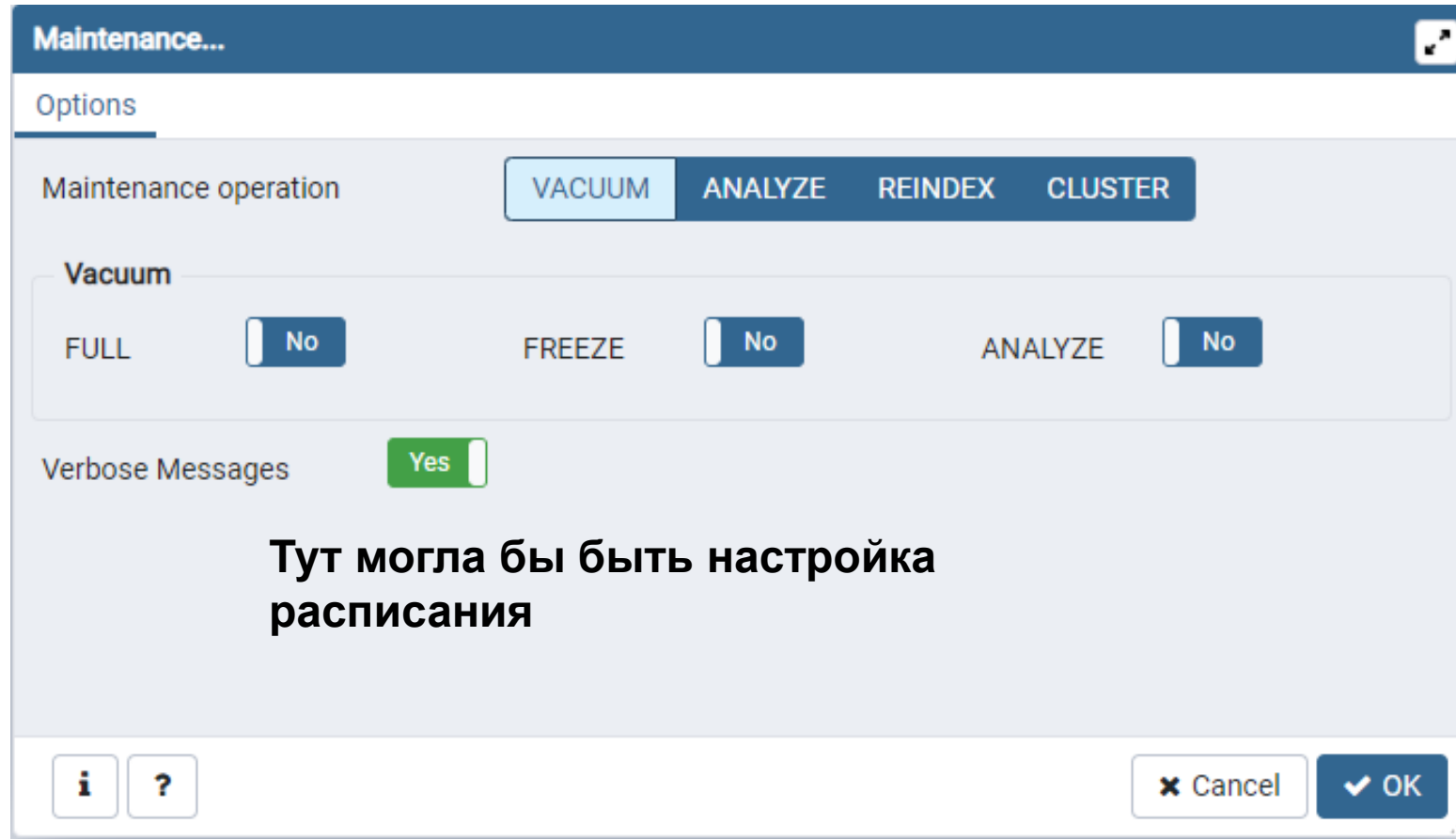
# PGAdmin — десктопное приложение



**PGAdmin3 — 1,5 сек**

**PGAdmin4 — 5 минут**

# PGAdmin — шедуллер обслуживания



# pg\_probackup

## Минимальная настройка

Данный сценарий иллюстрирует настройку для выполнения полного и разностного копирования.

### 1. Настройте подключение через SSH с backup\_host к postgres\_host:

```
[backupman@backup_host] ssh-copy-id postgres@postgres_host
```

### 2. Настройте кластер Postgres Pro.

Из соображений безопасности для выполнения резервного копирования рекомендуется использовать отдельную базу данных.

```
postgres=#  
CREATE DATABASE backupdb;
```

Подключитесь к базе backupdb, создайте роль probackup и дайте ей следующие права:

```
backupdb=#  
BEGIN;  
CREATE ROLE probackup WITH LOGIN REPLICATION;  
GRANT USAGE ON SCHEMA pg_catalog TO probackup;  
GRANT EXECUTE ON FUNCTION pg_catalog.current_setting(text) TO probackup;  
GRANT EXECUTE ON FUNCTION pg_catalog.pg_is_in_recovery() TO probackup;  
GRANT EXECUTE ON FUNCTION pg_catalog.pg_start_backup(text, boolean, boolean) TO probackup;  
GRANT EXECUTE ON FUNCTION pg_catalog.pg_stop_backup(boolean, boolean) TO probackup;  
GRANT EXECUTE ON FUNCTION pg_catalog.pg_create_restore_point(text) TO probackup;  
GRANT EXECUTE ON FUNCTION pg_catalog.pg_switch_wal() TO probackup;  
GRANT EXECUTE ON FUNCTION pg_catalog.pg_last_wal_replay_lsn() TO probackup;  
GRANT EXECUTE ON FUNCTION pg_catalog.txid_current() TO probackup;  
GRANT EXECUTE ON FUNCTION pg_catalog.txid_current_snapshot() TO probackup;  
GRANT EXECUTE ON FUNCTION pg_catalog.txid_snapshot_xmax(txid_snapshot) TO probackup;  
GRANT EXECUTE ON FUNCTION pg_catalog.pg_control_checkpoint() TO probackup;  
COMMIT;
```

### 3. Проинициализируйте каталог резервных копий:

```
[backupman@backup_host]$ pg_probackup-11 init -B /mnt/backups  
INFO: Backup catalog '/mnt/backups' successfully initied
```

# pg\_probackup

## 4. Добавьте экземпляр pg-11 в каталог резервных копий:

```
[backupman@backup_host]$ pg_probackup-11 add-instance -B /mnt/backups --instance 'pg-11' --remote-host=postgres_host --remote-user=postgres -D /var/lib/postgresql/11/main
INFO: Instance 'node' successfully initied
```

## 5. Сделайте полную резервную копию:

```
[backupman@backup_host] pg_probackup-11 backup -B /mnt/backups --instance 'pg-11' -b FULL --stream --remote-host=postgres_host --remote-user=postgres -U probackup -d backup
INFO: Backup start, pg_probackup version: 2.2.0, instance: node, backup ID: PZ7YK2, backup mode: FULL, wal mode: STREAM, remote: true, compress-algorithm: none, compress-lev
INFO: Start transferring data files
INFO: Data files are transferred
INFO: wait for pg_stop_backup()
INFO: pg_stop_backup() successfully executed
INFO: Validating backup PZ7YK2
INFO: Backup PZ7YK2 data files are valid
INFO: Backup PZ7YK2 resident size: 196MB
INFO: Backup PZ7YK2 completed
```

## 6. Взгляните на содержимое каталога копий:

```
[backupman@backup_host] pg_probackup-11 backup -B /mnt/backups --instance 'pg-11'
```

```
BACKUP INSTANCE 'pg-11'
```

```
=====
Instance  Version  ID      Recovery Time      Mode  WAL Mode  TLI  Time  Data  WAL  Zratio  Start LSN  Stop LSN  Status
=====
node      11        PZ7YK2  2019-10-11 19:45:45+03  FULL  STREAM   1/0   11s  180MB 16MB   1.00   0/3C000028  0/3C000198  OK
=====
```



# pg\_probackup

## 7. Сделайте инкрементальную резервную копию в режиме DELTA:

```
[backupman@backup_host] pg_probackup-11 backup -B /mnt/backups --instance 'pg-11' -b delta --stream --remote-host=postgres_host --remote-user=postgres -U probackup -d backupdb
INFO: Backup start, pg_probackup version: 2.2.0, instance: node, backup ID: PZ7YMP, backup mode: DELTA, wal mode: STREAM, remote: true, compress-algorithm: none, compress-level: 0
INFO: Parent backup: PZ7YK2
INFO: Start transferring data files
INFO: Data files are transferred
INFO: wait for pg_stop_backup()
INFO: pg_stop_backup() successfully executed
INFO: Validating backup PZ7YMP
INFO: Backup PZ7YMP data files are valid
INFO: Backup PZ7YMP resident size: 32MB
INFO: Backup PZ7YMP completed
```

## 8. Добавьте несколько параметров в файл конфигурации pg\_probackup, чтобы их не надо было задавать в командной строке:

```
[backupman@backup_host] pg_probackup-11 set-config -B /mnt/backups --instance 'pg-11' --remote-host=postgres_host --remote-user=postgres -U probackup -d backupdb
```

## 9. Сделайте ещё одну инкрементальную копию в режиме DELTA, опустив некоторые из предыдущих параметров:

```
[backupman@backup_host] pg_probackup-11 backup -B /mnt/backups --instance 'pg-11' -b delta --stream
INFO: Backup start, pg_probackup version: 2.2.0, instance: node, backup ID: PZ7YR5, backup mode: DELTA, wal mode: STREAM, remote: true, compress-algorithm: none, compress-level: 0
INFO: Parent backup: PZ7YMP
INFO: Start transferring data files
INFO: Data files are transferred
INFO: wait for pg_stop_backup()
INFO: pg_stop_backup() successfully executed
INFO: Validating backup PZ7YR5
INFO: Backup PZ7YR5 data files are valid
INFO: Backup PZ7YR5 resident size: 32MB
INFO: Backup PZ7YR5 completed
```

# pg\_probackup

## 10. Взгляните на конфигурацию экземпляра:

```
[backupman@backup_host] pg_probackup-11 show-config -B /mnt/backups --instance 'pg-11'

# Backup instance information
pgdata = /var/lib/postgresql/11/main
system-identifier = 6746586934060931492
xlog-seg-size = 16777216
# Connection parameters
pgdatabase = backupdb
pghost = postgres_host
pguser = probackup
# Replica parameters
replica-timeout = 5min
# Archive parameters
archive-timeout = 5min
# Logging parameters
log-level-console = INFO
log-level-file = OFF
log-filename = pg_probackup.log
log-rotation-size = 0
log-rotation-age = 0
# Retention parameters
retention-redundancy = 0
retention-window = 0
wal-depth = 0
# Compression parameters
compress-algorithm = none
compress-level = 1
# Remote access parameters
remote-proto = ssh
remote-host = postgres_host
```





Заметьте, что параметры, не переопределённые командой `set-config`, имеют значения по умолчанию.

## 11. Взгляните на содержимое каталога копий:

```
[backupman@backup_host] pg_probackup-11 backup -B /mnt/backups --instance 'pg-11'
```

Instance	Version	ID	Recovery Time	Mode	WAL Mode	TLI	Time	Data	WAL	Zratio	Start LSN	Stop LSN	Status
node	11	PZ7YR5	2019-10-11 19:49:56+03	DELTA	STREAM	1/1	10s	112kB	32MB	1.00	0/41000028	0/41000160	OK
node	11	PZ7YMP	2019-10-11 19:47:16+03	DELTA	STREAM	1/1	10s	376kB	32MB	1.00	0/3E000028	0/3F0000B8	OK
node	11	PZ7YK2	2019-10-11 19:45:45+03	FULL	STREAM	1/0	11s	180MB	16MB	1.00	0/3C000028	0/3C000198	OK





# pg\_probackup — графический интерфейс

Вложенный план	Описание	Расписание		
ВложенныйПлан_1	ВложенныйПлан_1	Выполняется раз в день в 20:00:00. Расписа...		
ВложенныйПлан_2	ВложенныйПлан_2	Выполняется раз в день, каждые 2 ч между...		

## Задача "Резервное копирование базы данных"



Создать резервную копию базы данных на Соединение с локальным сервером  
Базы данных Все пользовательские базы данных  
Тип: Полная  
Добавить к существующему  
Назначение: Диск  
Сжимать резервные копии (On)

Вложенный план	Описание	Расписание		
ВложенныйПлан_1	ВложенныйПлан_1	Выполняется раз в день в 20:00:00. Расписа...		
ВложенныйПлан_2	ВложенныйПлан_2	Выполняется раз в день, каждые 2 ч между...		

## Задача "Резервное копирование базы данных"



Создать резервную копию базы данных на Соединение с локальным сервером  
Базы данных Все пользовательские базы данных  
Тип: Разностная  
Добавить к существующему  
Назначение: Диск  
Сжимать резервные копии (On)

# pgpro\_scheduler

Имя:  Задания в расписании

Тип расписания:   Включено

Однократное выполнение

Дата:  Время:

Частота

Выполняется:

Повторяется каждые:  дн.

Сколько раз в день

Однократное задание:

Выполняется кажд.:

Продолжительность

Дата начала:

Дата окончания:

Без даты окончания:

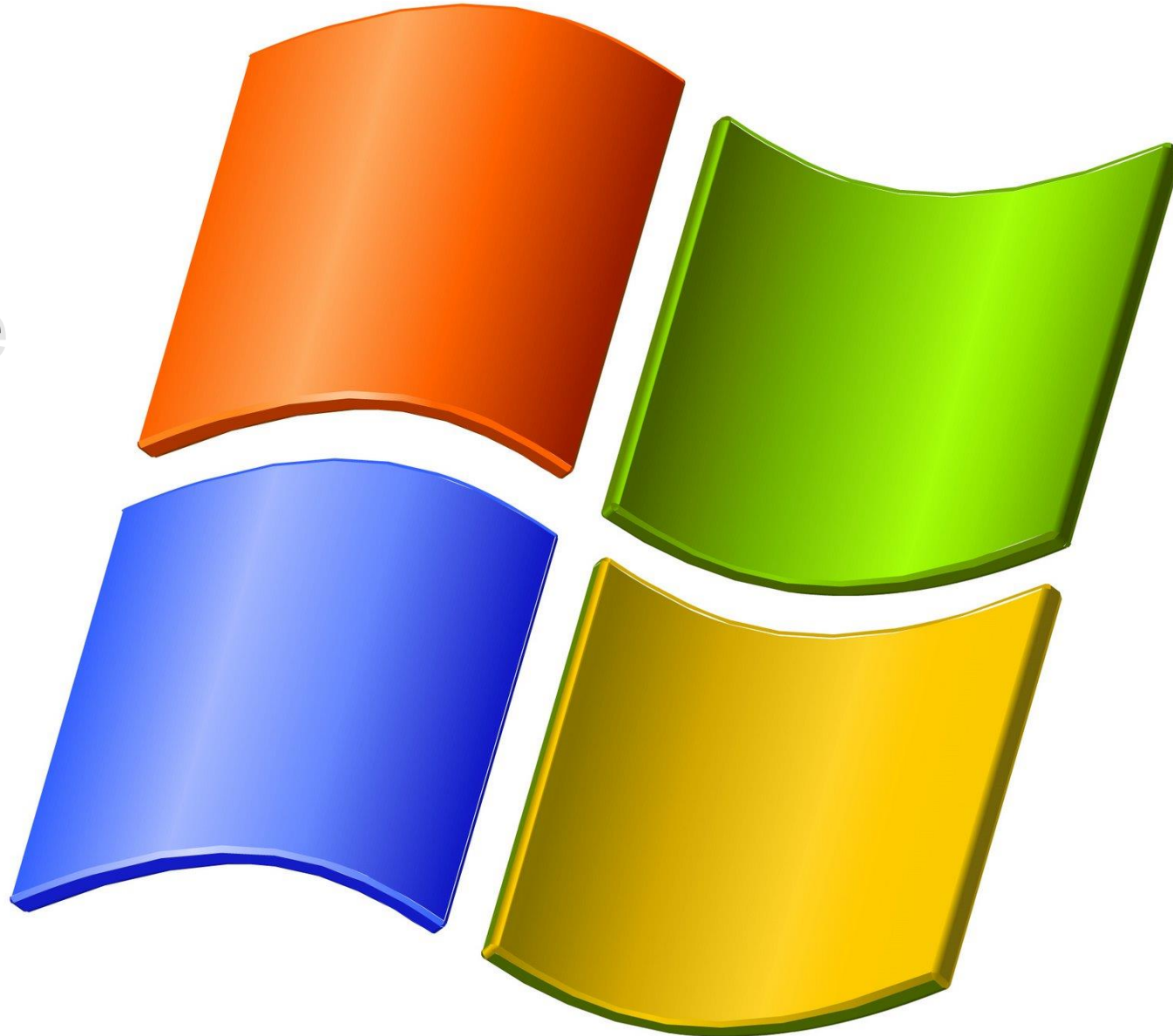
Сводка

Описание:

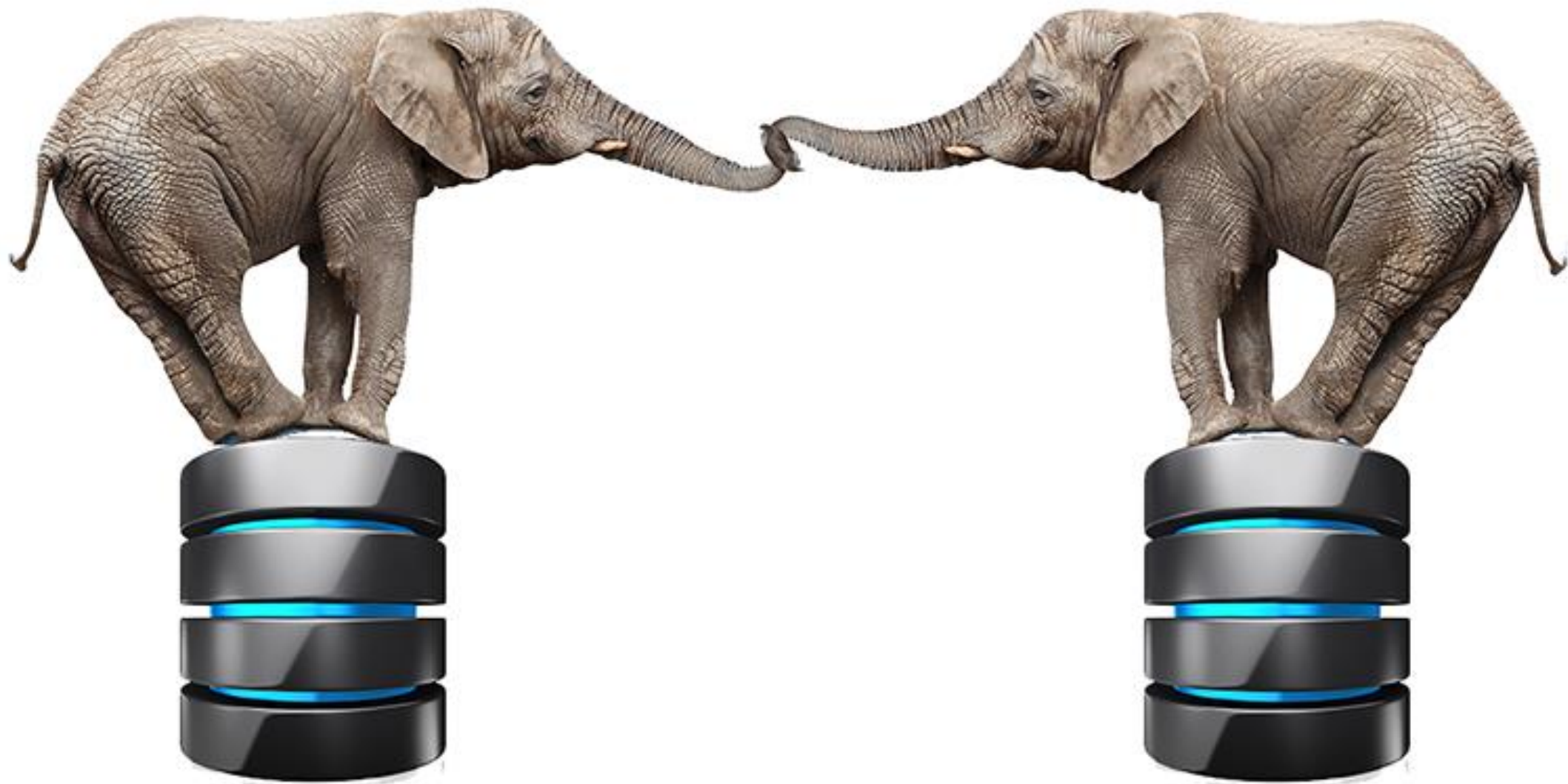
# Pgpro Enterprise

**pg\_dump**

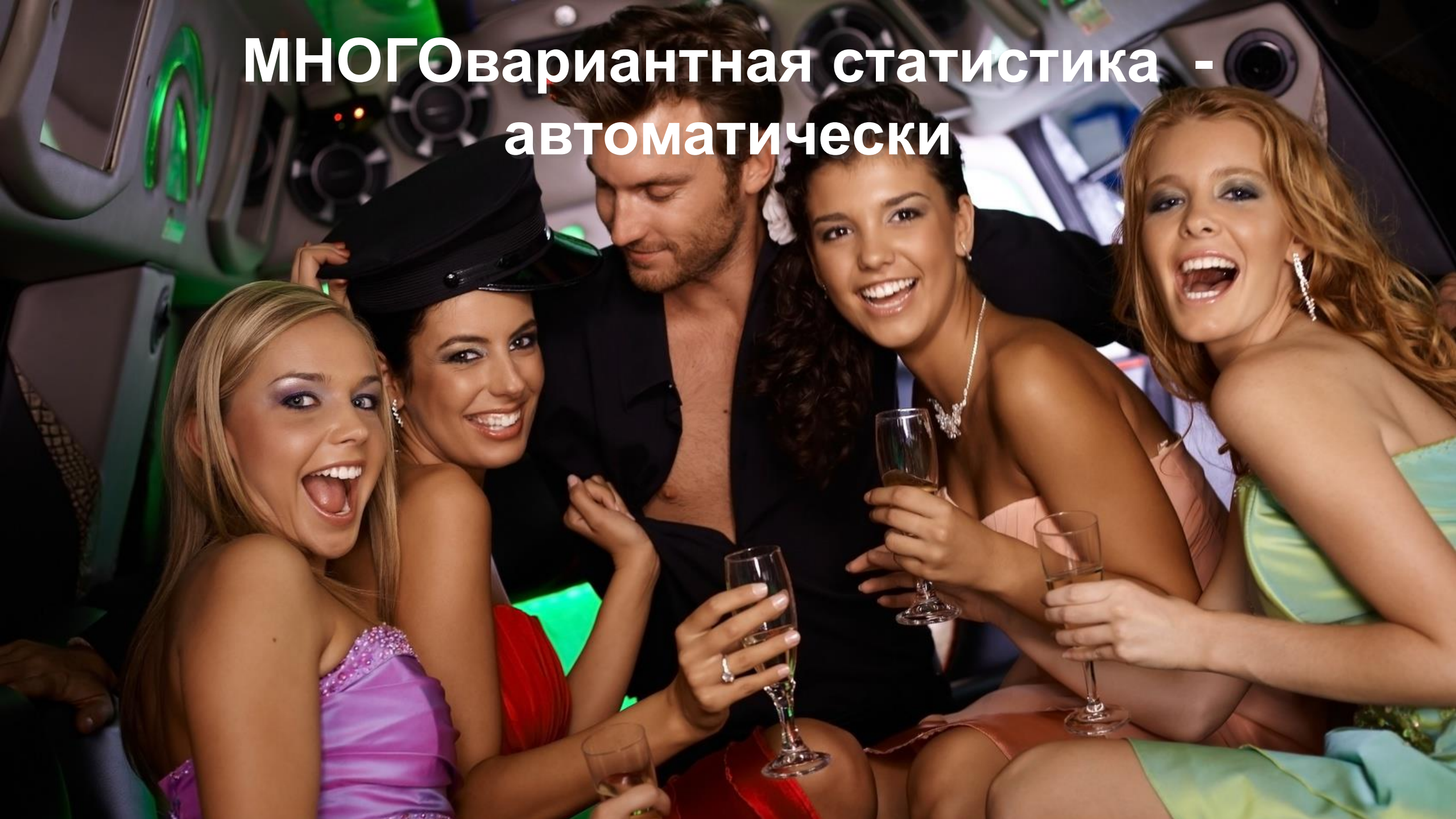
**pg\_restore**



# Кластер из коробки



# МНОГОВАРИАНТНАЯ СТАТИСТИКА - АВТОМАТИЧЕСКИ



# Tablespace и реструктуризация 1С



**TableName\_ng**



# Полнотекстовый поиск на уровне БД



**Like → contains**

**pg\_dump — даёшь больше 1ГБ!**



**--db-include=имя\_базы  
pg\_probackup restore только нужной БД**





PGConf.Russia

# С заботой о Вас и Вашей 1С

Дорошкевич АНТОН

ИнфоСофт

<https://www.facebook.com/doroshkevich.anton>

[a.doroshkevich@is1c.ru](mailto:a.doroshkevich@is1c.ru)

<https://is1c.ru>

