



**deepseek**



New chat

Today

Копирование базы данных с FDW

...

Мониторинг скорости чтения/записи nvme0n1

...

Yesterday

Расчет эффективности индексов в PostgreSQL

...

Настройка параметров CREATE TABLE в PostgreSQL

...

7 Days



Структура и использование pg\_largeobject

...

Почему отображаются только индексы и TOAST

...

Исправление ошибки в SQL-запросе

...

PostgreSQL Table Size Statistics Script Fix

...

Объединение сводных таблиц в SQL

...

SQL Query for Database Table Statistics

...

для списка баз данных "casino\_bo

...

Переписанные заголовки для таблиц БД

...

SQL Query Error: Column Not Found

...

Забыть предыдущую переписку и начать заново

...

SQL Query Syntax Error Correction Guide

...

Изменение фона комментариев в DataGrip

...

Removing Quotes from CSV Output

...

Сравнение размеров таблиц в PostgreSQL

...

Сформирован список команд подключения к БД

...

Продолжить для списка баз данных

...

PL/pgSQL и переключение баз данных

...

postgresql

...

30 Days

Настройка DataGrip: отключение подсветки комментариев

...

PostgreSQL 17 psql Commands Reference Guide

...

Запрос о дата-центре для ответа

...

Анализ параллельного выполнения в PostgreSQL

...

Преобразование скрипта в CTE без изменений

...

New chat

...

Исправление SQL-запроса для таблиц

...

Find Largest Tables in PostgreSQL Databases

...

Получение привилегий в PostgreSQL

...

Создание функции для базы данных PostgreSQL

...

PostgreSQL Script Syntax Error Fix

...

Исправление скрипта для переноса баз данных

...

Миграция данных между PostgreSQL через PL/pgSQL

...

Исправление синтаксической ошибки и упрощение кода

...

Исправление ошибки копирования данных SQL

...

Скрипт PostgreSQL для предоставления прав

...

Инициализация PostgreSQL без локалей

...

PL/pgSQL: ошибка dblink\_exes и исправление

...

Вызов удаленной функции через dblink в PL/pgSQL

...

Скрипт PL/pgSQL для синхронизации таблиц

...

Замена \*\*\* на JSONB в функции

...

Редактирование PL/pgSQL для вызова удаленной функции

...

Исправление ошибок в PL/pgSQL коде

...

Функция PL/pgSQL для интервалов времени

...

Экранирование одинарных кавычек в PL/pgSQL

...

PostgreSQL dblink pgbadger Execution Error Fix

...

PostgreSQL запрос с ошибками и исправлениями

...

Конструкция CASE в PL/pgSQL подробно

...

2025-03

Ограничения размера HTML-файла в браузерах

...

Проверка создания логов PostgreSQL

...

PostgreSQL log\_statement значение none объяснение

...

Проверка работы таблиц логирования PostgreSQL

...

Определение значения по умолчанию PostgreSQL

...

Запуск pgBadger для анализа PostgreSQL логов

...

Создание таблицы с динамическим пользователем

...

Git push rejected: resolve conflicts and pull

...

SQL-запрос с JOIN для объединения таблиц

...

Fixing SQL Unterminated String Error

...

Отключение двойных кавычек в Notepad++

...

PostgreSQL Ambiguous Column Error Resolution

...

Запуск cron для PostgreSQL процедуры

...

Исправление ошибок в коде PostgreSQL

...

Удаление записи в журналы и исключений

...

Упрощение скрипта PostgreSQL для роли

...

Создание процедуры для реиндексации индексов

...

Ошибка деления на ноль в SQL-запросе

...

PostgreSQL DBLINK Update Error Fix

...

SQL Syntax Error in DBLINK Query Fix

...

PL/pgSQL Syntax Error and Correction Guide

...

Определение доступа роли к базам данных

...



Get App **NEW**

My Profile



Копирование базы данных с FDW

Как с помощью FDW скопировать базу данных из примапленного сервера?



## Копирование базы данных с помощью FDW (Foreign Data Wrapper) из примапленного сервера

---

Для копирования базы данных с удаленного сервера с использованием FDW в PostgreSQL можно выполнить следующие шаги:

### 1. Установка расширения FDW

---

sql



Copy



Download

```
CREATE EXTENSION postgres_fdw;
```

### 2. Создание сервера-источника

---

sql



Copy



Download

```
CREATE SERVER remote_server  
FOREIGN DATA WRAPPER postgres_fdw  
OPTIONS (host 'имя_хоста_или_ip', port '5432', dbname 'имя_базы_данных');
```

### 3. Создание mapping пользователя

---

sql



Copy



Download

```
CREATE USER MAPPING FOR текущий_пользователь
SERVER remote_server
OPTIONS (user 'удаленный_пользователь', password 'пароль');
```

## 4. Создание внешних таблиц

---

### Вариант А: Импорт всей схемы

---

sql



Copy



Download

```
IMPORT FOREIGN SCHEMA public
FROM SERVER remote_server
INTO public;
```

### Вариант В: Импорт конкретных таблиц

---

sql



Copy



Download

```
CREATE FOREIGN TABLE remote_table_name (
    id integer,
    column1 text,
    column2 timestamp
)
SERVER remote_server
OPTIONS (schema_name 'public', table_name 'actual_table_name');
```

## 5. Копирование данных

---

### Способ 1: Создание новой таблицы и копирование

---

sql



Copy



Download

```
CREATE TABLE local_table AS  
SELECT * FROM remote_table_name;
```

## Способ 2: Использование pg\_dump/pg\_restore

---

Если у вас есть доступ к серверу, можно использовать:

bash



Copy



Download

```
pg_dump -h удаленный_сервер -U пользователь -d база_данных > dump.sql  
psql -h локальный_сервер -U пользователь -d целевая_база < dump.sql
```

## Способ 3: Использование dblink (альтернатива FDW)

---

sql



Copy



Download

```
-- Создание подключения  
SELECT dblink_connect('myconn', 'dbname=remote_db user=remote_user password=secret  
host=remote_host');  
  
-- Копирование данных  
CREATE TABLE local_table AS  
SELECT * FROM dblink('myconn', 'SELECT * FROM remote_table') AS t(id int, col1 text,  
col2 date);  
  
-- Закрытие подключения  
SELECT dblink_disconnect('myconn');
```

## Важные замечания:

---

1. Для больших баз данных копирование через FDW может быть медленным
2. Убедитесь, что у вас есть необходимые права на удаленном сервере
3. Для сложных структур (индексы, триггеры, ограничения) может потребоваться дополнительная настройка



4. Рассмотрите использование инструментов типа `pg_dump/pg_restore` для полного копирования базы

Если вам нужно скопировать всю базу данных с сохранением всех объектов, лучше использовать `pg_dump` напрямую с удаленного сервера.



Как? Мне нужно скопировать всю базу данных с сохранением всех объектов, `pg_dump` напрямую с удаленного сервера.



## Копирование всей базы данных с удаленного PostgreSQL сервера с помощью `pg_dump`

---

Для полного копирования базы данных со всеми объектами (таблицами, индексами, триггерами, функциями, схемами и т.д.) лучше всего использовать утилиту `pg_dump` напрямую.

### 1. Базовый синтаксис `pg_dump`

---

bash



Copy



Download

```
pg_dump -h <remote_host> -p <port> -U <username> -d <database_name> -F c -f  
<output_file> -v
```

Где:

- `-h` - хост удаленного сервера
- `-p` - порт (по умолчанию 5432)
- `-U` - имя пользователя
- `-d` - имя базы данных
- `-F c` - формат вывода (custom, рекомендуется для полного копирования)

- **-f** - имя выходного файла
- **-v** - verbose режим (опционально)

## 2. Пример полного копирования

---

bash



Copy



Download

```
pg_dump -h remote.example.com -p 5432 -U postgres -d my_database -F c -f  
my_database.dump -v
```

После выполнения команды вам будет предложено ввести пароль.

## 3. Восстановление базы данных на целевом сервере

---

bash



Copy



Download

```
pg_restore -h localhost -p 5432 -U postgres -d new_database -C -v my_database.dump
```

Опции **pg\_restore**:

- **-C** - создает базу данных перед восстановлением
- Остальные параметры аналогичны **pg\_dump**

## 4. Альтернативные варианты

---

### Копирование напрямую между серверами (без промежуточного файла)

---

bash



Copy



Download

```
pg_dump -h remote_host -U remote_user -d remote_db | psql -h local_host -U local_user  
-d local_db
```

## Копирование с сжатием

---

bash



Copy



Download

```
pg_dump -h remote_host -U remote_user -d remote_db | gzip > remote_db_dump.gz  
# Для восстановления:  
gunzip -c remote_db_dump.gz | psql -h local_host -U local_user -d local_db
```

## 5. Дополнительные полезные опции pg\_dump

---

- **--schema-only** - только схема, без данных
- **--data-only** - только данные, без схемы
- **--no-owner** - не сохранять владельцев объектов
- **--no-privileges** - не сохранять права доступа
- **--clean** - добавить команды DROP перед CREATE
- **--create** - добавить команду CREATE DATABASE

## 6. Для очень больших баз данных

---

Для больших баз можно использовать параллельный дамп:

bash



Copy



Download

```
pg_dump -h remote_host -U remote_user -d remote_db -j 4 -F d -f dump_dir
```

Где:

- **-j 4** - количество параллельных задач
- **-F d** - формат директории (создает папку с файлами)

Для восстановления параллельного дампа:

bash



Copy



Download

```
pg_restore -h local_host -U local_user -d local_db -j 4 -C dump_dir
```

## Важные замечания:

---

1. Убедитесь, что у пользователя есть права на чтение всей базы данных
2. Для больших баз может потребоваться значительное время и место на диске
3. Рекомендуется использовать формат "custom" (**-F c**) для полного сохранения всех объектов
4. При копировании между разными версиями PostgreSQL могут быть проблемы с совместимостью



`pg_dump -h remote_host -U remote_user -d remote_db -j 4 -F d -f dump_dir` для  
Копирование напрямую между серверами (без промежуточного файла)



Чтобы выполнить прямое копирование базы данных между серверами **без промежуточного файла**, используя параллельный режим (**-j N**), можно воспользоваться комбинацией **pg\_dump** и **pg\_restore** через пайп (pipe).

## 1. Прямое копирование с параллельным дампом и восстановлением

---

```
bash
```



Copy



Download

```
pg_dump -h remote_host -U remote_user -d remote_db -j 4 -F c | \
pg_restore -h local_host -U local_user -d local_db -j 4 --clean --if-exists -C
```

### Опции:

---

- **-F c** – формат **custom** (обязателен для параллельного режима **-j** в **pg\_dump**).

- `-j 4` – количество потоков (можно увеличить, если сервер мощный).
- `--clean` – удаляет существующие объекты перед восстановлением.
- `--if-exists` – добавляет `IF EXISTS` к командам DROP (избегает ошибок, если объекты не существуют).
- `-C` – создает базу данных перед восстановлением (если её нет).

---

## 2. Если нужно сохранить владельцев и привилегии

---

bash



Copy



Download

```
pg_dump -h remote_host -U remote_user -d remote_db -F c -j 4 | \  
pg_restore -h local_host -U local_user -d local_db -j 4 -C --no-owner --no-privileges
```

### Когда использовать `--no-owner` и `--no-privileges`?

---

- Если пользователи на **удаленном** и **локальном** серверах разные, лучше отключить сохранение владельцев (`--no-owner`), иначе восстановление может завершиться ошибкой.
- `--no-privileges` отключает копирование прав доступа (GRANT/REVOKE).

---

## 3. Если база уже существует и нужно просто обновить данные

---

bash



Copy



Download

```
pg_dump -h remote_host -U remote_user -d remote_db -F c -j 4 | \  
pg_restore -h local_host -U local_user -d local_db -j 4 --clean --if-exists
```

(без `-C`, если база уже создана).

---

## 4. Альтернатива: `psql` + `pg_dump` (если `pg_restore` недоступен)

---

bash



Copy



Download

```
pg_dump -h remote_host -U remote_user -d remote_db -F p -j 4 | \  
psql -h local_host -U local_user -d local_db
```

- **-F p** (plain-text SQL) – работает, но медленнее, чем бинарный формат (**-F c**).
- **Не поддерживает -j в psql**, поэтому распараллеливание только на этапе дампа.

---

## Ошибки и решения

1. **"Role does not exist"** → Используйте **--no-owner**.
2. **"Database already exists"** → Уберите **-c** или сначала удалите старую БД.
3. **Нет доступа к pg\_dump на удаленном сервере?** → Используйте SSH-туннель:

```
bash
```



Copy



Download

```
ssh user@remote_host "pg_dump -U remote_user -d remote_db -F c -j 4" | \  
pg_restore -h localhost -U local_user -d local_db -j 4 -C
```

---

## Вывод

Лучший способ для **быстрого прямого копирования** – это:

```
bash
```



Copy



Download

```
pg_dump -h remote_host -U remote_user -d remote_db -F c -j 4 | \  
pg_restore -h local_host -U local_user -d local_db -j 4 -C --clean --if-exists
```

Этот метод **не сохраняет временный файл**, работает быстро (благодаря параллельности) и подходит для больших БД.





↻ New chat



DeepThink (R1)



Search



AI-generated, for reference only