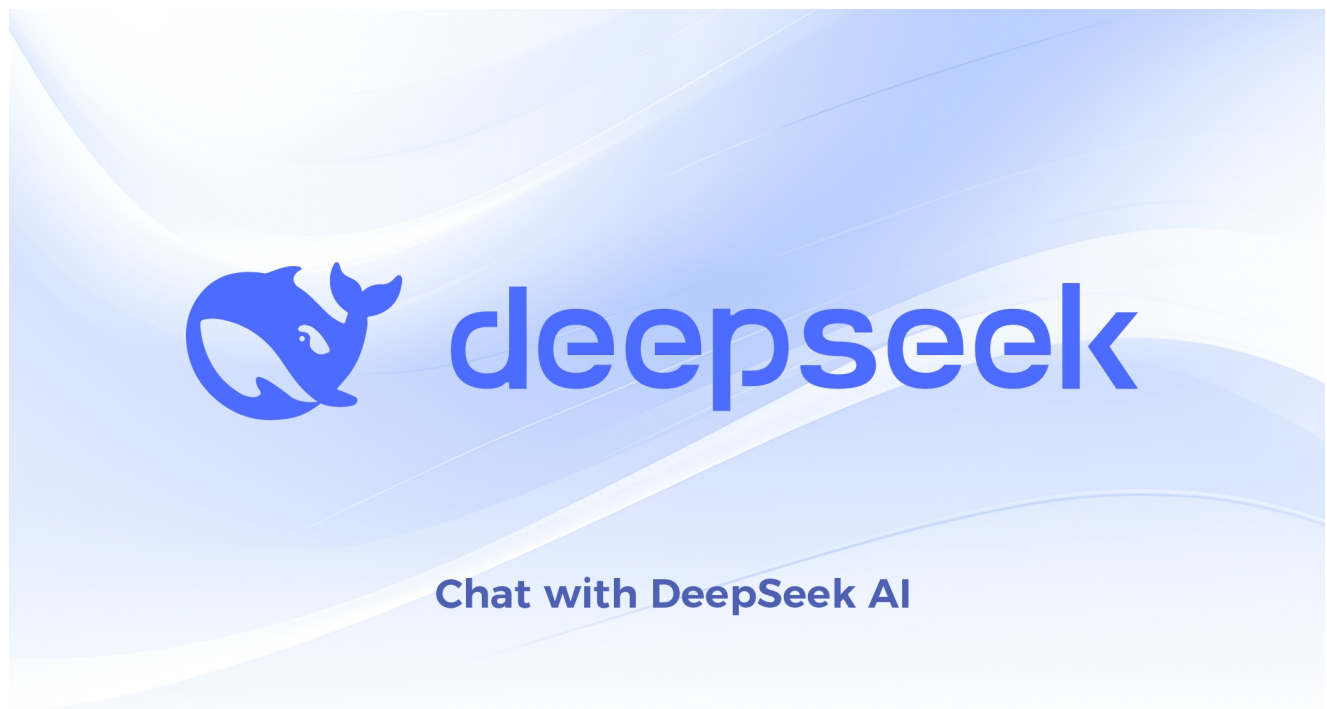


DeepSeek - Into the Unknown

 chat.deepseek.com/a/chat/s/e3dbbc1c-8117-4e00-8775-262dde9c3010



Создание и загрузка дампов баз данных

...

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

Создание и загрузка дампов баз данных



Каталог для дампов: /home/temp Шаг 1. Создание дампа всех таблиц, кроме users, user_project, user_groups, groups export PGPASSWORD="d56f2c4eFDnofa1w" pg_dump \ --host=prd-chat-pg-02.maxbit.private \ --port=5434 \ --username=gtimofeyev \ -- dbname=1go_mbss_master \ --format=custom \ --no-owner \ --no-privileges \ --verbose \ -- exclude-table=users \ --exclude-table=user_project \ --exclude-table=user_groups \ -- exclude-table=groups \ --file=/home/temp/full.dump Шаг 2. Дамп только таблиц users, user_project, user_groups, groups в SQL-формате pg_dump \ --host=prd-chat-pg-02.maxbit.private \ --port=5434 \ --username=gtimofeyev \ --dbname=1go_mbss_master \ -- format=plain \ --data-only \ --table=users \ --table=user_project \ --table=user_groups \ -- table=groups \ --no-owner \ --no-privileges \ --verbose \ > /home/temp/partial.sql Шаг 3. Модификация SQL-дампа: ON CONFLICT DO NOTHING sed -i -E 's/^INSERT INTO ([^]+) /INSERT INTO \1 ON CONFLICT DO NOTHING /' /home/temp/partial.sql Шаг 4. Очистка базы local от всех таблиц (кроме user-таблиц) DO \$\$ DECLARE r RECORD; BEGIN FOR r IN SELECT tablename FROM pg_tables WHERE schemaname = 'public' AND tablename NOT IN ('users', 'user_project', 'user_groups', 'groups') LOOP EXECUTE format('TRUNCATE TABLE public.%I CASCADE', r.tablename); END LOOP; END \$\$; Шаг 5. Загрузка из дампа всех таблиц (кроме user-таблиц) Шаг 6. Загрузка из дампа user-таблиц Шаг 7. Очистка переменной пароля unset PGPASSWORD дополнить шаг 5 и шаг 6



Вот дополненные шаги 5 и 6 для вашего скрипта:

Шаг 5. Загрузка из дампа всех таблиц (кроме user-таблиц)

bash



Copy



Download

```
pg_restore \  
  --host=localhost \  
  --port=5432 \  
  --username=your_local_username \  
  --dbname=your_local_dbname \  
  --no-owner \  
  --no-privileges \  
  --verbose \  
  /home/temp/full.dump
```

Замените `your_local_username` и `your_local_dbname` на актуальные значения для вашей локальной базы данных.

Шаг 6. Загрузка из дампа user-таблиц

bash



Copy



Download

```
psql \  
  --host=localhost \  
  --port=5432 \  
  --username=your_local_username \  
  --dbname=your_local_dbname \  
  --file=/home/temp/partial.sql
```

Опять же, укажите правильные параметры для вашей локальной базы данных. Так как `partial.sql` уже был модифицирован (шаг 3) для добавления `ON CONFLICT DO NOTHING`, данные будут вставлены только для отсутствующих записей.

После выполнения этих шагов вы можете продолжить с шагом 7 (очистка переменной пароля).



В предоставленном мной сценарии выполнять `pg_dump` и `pg_restore` в каком сервере?



В вашем сценарии **pg_dump** выполняется на продакшн-сервере PostgreSQL (**prd-chat-pg-02.maxbit.private:5434**), а **pg_restore** и **psql** — на локальном сервере PostgreSQL (**localhost:5432**).

Разбор команд:

Шаг 1 и 2 (**pg_dump**)

bash



Copy



Download

```
pg_dump \  
  --host=prd-chat-pg-02.maxbit.private \  
  --port=5434 \  
  --username=gtimofeyev \  
  --dbname=1go_mbss_master \  
  --file=/home/temp/full.dump
```

Продакшн-сервер
Порт продакшн-сервера
Пользователь продакшн-БД
Продакшн-база данных
Дамп сохраняется локально

→ **Выполняется на продакшн-сервере**, но дамп сохраняется в **/home/temp** (предположительно, на машине, откуда запускается скрипт).

Шаг 5 (**pg_restore**) и Шаг 6 (**psql**)

bash



Copy



Download

```
pg_restore \  
  --host=localhost \  
  --port=5432 \  
  --username=your_local_username \  
  --dbname=your_local_dbname \  
  /home/temp/full.dump
```

Локальный сервер PostgreSQL
Стандартный порт PostgreSQL
Локальный пользователь
Локальная БД

→ **Выполняется на локальной машине**, куда предварительно скопированы дампы (**full.dump** и **partial.sql**).

Уточнения:

1. Где запускаются команды?

- Если скрипт выполняется **на продакшн-сервере**, то дампы создаются прямо на нём (в `/home/temp`).
- Если скрипт запускается **с вашей локальной машины**, то продакшн-сервер должен быть доступен по сети, а у пользователя `gtimofeyev` должны быть права на чтение БД.

2. Куда загружаются данные (`pg_restore`)?

В локальную БД (`localhost:5432`), например, в тестовое окружение или dev-среду.

3. Безопасность

- Пароль (`PGPASSWORD`) передаётся в переменной окружения, что безопаснее, чем в командной строке.
- После завершения работы пароль очищается (`unset PGPASSWORD`).

Рекомендации:

- Если `/home/temp` находится на продакшн-сервере, убедитесь, что дампы скопированы на локальную машину перед `pg_restore`.
- Для больших БД используйте `-j N` (многопоточность) в `pg_restore` для ускорения загрузки.
- Проверьте, что локальная БД **уже создана** и пользователь имеет права на запись.

Пример копирования дампов с продакшна на локальную машину (если нужно):

```
bash
```



Copy



Download

```
scp user@prd-chat-pg-02.maxbit.private:/home/temp/{full.dump,partial.sql} /home/temp/
```



Для больших БД используйте `-j N` (многопоточность) в `pg_restore` для ускорения загрузки и выгрузки?



Да, для **ускорения работы с большими базами данных** в `pg_restore` и `pg_dump` можно использовать **многопоточность** (`-j N`). Однако есть важные нюансы:

1. Для `pg_restore` (загрузка дампа)

Как использовать:

bash



Copy



Download

```
pg_restore \  
  --host=localhost \  
  --port=5432 \  
  --username=your_user \  
  --dbname=your_db \  
  --jobs=4 \  
  --verbose \  
/home/temp/full.dump
```

Число потоков (например, 4)

Когда применять:

- Работает **только с форматом `custom` (`-Fc`)** (который вы используете в шаге 1).
- Ускоряет загрузку **таблиц, индексов и `constraints`**.

Ограничения:

- Не поддерживается для **SQL-формата** (ваш `partial.sql` в шаге 2).
- Если БД маленькая, многопоточность может дать **обратный эффект** из-за накладных расходов.

2. Для `pg_dump` (выгрузка дампа)

Как использовать:

bash



Copy



Download

```
pg_dump \  
--host=prd-chat-pg-02.maxbit.private \  
--port=5434 \  
--username=gtimofeyev \  
--dbname=1go_mbss_master \  
--format=custom \           # Обязательно для -j  
--jobs=4 \                  # Число потоков  
--file=/home/temp/full.dump
```

Когда применять:

- Параллельно выгружаются **только таблицы**. Индексы, триггеры и т.д. сохраняются в одном потоке.
- Максимальный эффект — на **больших таблицах** (например, таблицы с миллионами строк).

Ограничения:

- Не работает с `--format=plain` (SQL-дампы, как ваш `partial.sql`).
- Требуется **отдельного подключения к БД для каждого потока** (убедитесь, что `max_connections` в PostgreSQL позволяет это).

3. Оптимальное число потоков (-j N)

- **Рекомендуется: $N = \text{число ядер CPU} - 1$** (например, для 4-ядерного CPU используйте `-j 3`).
- **Слишком большое N** может перегрузить сервер или сеть.

Итоговые доработки вашего скрипта

Для шага 1 (`pg_dump`):

```
bash
```



Copy



Download

```
pg_dump \  
  --host=prd-chat-pg-02.maxbit.private \  
  --port=5434 \  
  --username=gtimofeyev \  
  --dbname=1go_mbss_master \  
  --format=custom \  
  --jobs=4 \                # Добавлено  
  --no-owner \  
  --no-privileges \  
  --exclude-table=users \  
  --exclude-table=user_project \  
  --exclude-table=user_groups \  
  --exclude-table=groups \  
  --file=/home/temp/full.dump
```

Для шага 5 (pg_restore):

bash



Copy



Download

```
pg_restore \  
  --host=localhost \  
  --port=5432 \  
  --username=your_local_user \  
  --dbname=your_local_db \  
  --jobs=4 \                # Добавлено  
  --no-owner \  
  --no-privileges \  
  --verbose \  
  /home/temp/full.dump
```

Важно!

1. Тестируйте на тестовом окружении, особенно если БД критична.
2. Мониторьте нагрузку на сервер при использовании `-j` (например, через `htop` или `pg_stat_activity`).
3. Для **SQL-дампа** (`partial.sql`) многопоточность невозможна — используйте только для `custom/directory`-форматов.

Если у вас есть **особо большие таблицы**, можно также рассмотреть:

- Разделение дампа на части с `pg_dump --table=...`

- Сжатие на лету с `pigz/zstd` (например, `pg_dump | pigz > dump.gz`).



 New chat



DeepThink (R1)



Search



AI-generated, for reference only