PostgreSQL

- Установка и управление PostgreSQL
- Установка pgAdmin

????????? ? ??????????
PostgreSQL

????????? ? ???????????

??? 1: ????????? ???????

```
sudo apt update && sudo apt install -y postgresql postgresql-contrib
```

??? 2: ??????????? ? ??????? psql

PostgreSQL создает пользователя `postgres`. Для подключения к СУБД нужно сначала переключиться на этого системного пользователя.

```
sudo -u postgres psql
```

??? 3: ????????? ?????? ??? ???????????? postgres

По умолчанию у пользователя `postgres` нет пароля. Зададим его для безопасности.

Сначала войдите в psql (команда выше), затем выполните:

```
\password postgres
```

После этого выйдите из консоли командой `\\q`.

?????????? ?????? ??????

Все команды выполняются в консоли `psql`.

???????? ?? (`CREATE DATABASE`)

```
CREATE DATABASE my_new_app;
```

???????? ???? ??

```
\l
```

??????????? ? ?????? ??

```
\c my_new_app
```

???????? ?? (`DROP DATABASE`)

Внимание! Команда необратима.

```
DROP DATABASE my_new_app;
```

??????????? ?????? (???????????????)

Все команды выполняются в консоли `psql`.

???????? ???? (???????????)

`LOGIN` означает, что роль может входить в систему. `PASSWORD` задает пароль.

```
CREATE ROLE newuser WITH LOGIN PASSWORD 'a_very_strong_password';
```

????????????? ????

Дает пользователю все права на определенную базу данных.

```
GRANT ALL PRIVILEGES ON DATABASE my_new_app TO newuser;
```

?????? ? ??????? (SQL ? psql-???????)

Основные команды для взаимодействия с таблицами и данными внутри базы. Команды, начинающиеся с `\\`, являются внутренними командами клиента `psql`.

???????? ?????? ? ?? ?????????

После подключения к базе (`\\c my_new_app`) можно посмотреть список таблиц:

```
\dt
```

Посмотреть структуру конкретной таблицы (столбцы, типы, индексы):

```
\d users
```

????? ? ????????? ??????

Найти всех пользователей, зарегистрированных после определенной даты:

```
SELECT * FROM users WHERE registration_date > '2023-01-01';
```

Изменить статус пользователя с ID 123 на 'inactive':

```
UPDATE users SET status = 'inactive' WHERE id = 123;
```

?????????? ??????????? ??????? ? ????

`psql` имеет очень удобную команду `\\copy`, которая выполняет SQL-запрос на сервере и сохраняет результат в файл **на вашем локальном компьютере**, от имени вашего локального пользователя.

```
\copy (SELECT id, email FROM users WHERE is_active = true) TO '/home/myuser/active_users.csv'
WITH CSV HEADER;
```

???????? ? ?????? ??????

В отличие от MySQL, в PostgreSQL нет прямого аналога `REPAIR TABLE`. Благодаря своей транзакционной архитектуре (MVCC и WAL), повреждение данных на уровне страниц происходит крайне редко. Основной инструмент обслуживания — это `REINDEX`.

??????????? ???????? (`REINDEX`)

Если вы подозреваете, что индексы таблицы "раздулись" или повреждены (что может приводить к замедлению запросов), вы можете их перестроить. Эта операция безопасна.

Перестроить все индексы одной таблицы:

```
REINDEX TABLE my_table;
```

Перестроить все индексы в базе данных:

```
REINDEX DATABASE my_database_name;
```

Важно: В случае серьезного повреждения данных, которое не исправляется с помощью `REINDEX`, единственной правильной стратегией для PostgreSQL является восстановление из последней работоспособной резервной копии, созданной с помощью `pg_dump`.

????????? ???????????? ? ?????????????

Эти команды выполняются из обычной командной строки Linux, от имени пользователя `postgres`.

????????? ??????????? (Backup)

`pg_dump` создает файл бэкапа.

```
sudo -u postgres pg_dump my_database_name > backup.sql
```

????????????? (Restore)

Для восстановления сначала нужно создать пустую базу данных и назначить ей владельца, а затем импортировать данные.

1. Создаем пустую базу и назначаем владельца в `psql`:

```
CREATE DATABASE my_database_name OWNER newuser;
```

2. Импортируем данные в командной строке Linux:

```
sudo -u postgres psql my_database_name < backup.sql
```

????????? pgAdmin

??? Linux (Debian/Ubuntu)

pgAdmin для Linux устанавливается как веб-приложение. Вы будете получать доступ к нему через браузер.

Шаг 1: Установка зависимостей и добавление репозитория pgAdmin.

```
# Установка зависимостей
sudo apt update && sudo apt install -y curl gpg
# Добавление GPG ключа
curl -fsS https://www.pgadmin.org/static/packages_pgadmin_org.pub | sudo gpg --dearmor -o
/usr/share/keyrings/packages-pgadmin-org.gpg
# Добавление репозитория
sudo sh -c 'echo "deb [signed-by=/usr/share/keyrings/packages-pgadmin-org.gpg]
https://ftp.postgresql.org/pub/pgadmin/pgadmin4/apt/$(lsb_release -cs) pgadmin4 main" >
/etc/apt/sources.list.d/pgadmin4.list && apt update'
```

Шаг 2: Установка веб-версии.

```
sudo apt install -y pgadmin4-web
```

Шаг 3: Запуск скрипта начальной настройки. Он попросит вас создать email и пароль для входа в веб-интерфейс pgAdmin.

```
sudo /usr/pgadmin4/bin/setup-web.sh
```

После завершения вы сможете получить доступ к pgAdmin в браузере по адресу `http://ip-вашего-сервера/pgadmin4`.

??? Windows

Установка для Windows предельно проста — скачайте инсталлятор и следуйте инструкциям мастера установки.

????????? ??????? PostgreSQL ??? ?????????? ???????????

Чтобы pgAdmin (или любой другой клиент) мог подключиться к вашему серверу PostgreSQL по сети, необходимо выполнить два ключевых шага на самом сервере.

??? 1: ????????? ????????????? ??????? ???????????

По умолчанию PostgreSQL слушает подключения только с самого себя (`localhost`). Это нужно изменить.

Откройте основной конфигурационный файл (путь может отличаться в зависимости от версии, например, `/14/`):

```
sudo nano /etc/postgresql/14/main/postgresql.conf
```

Найдите строку (она может быть закомментирована):

```
#listen_addresses = 'localhost'
```

Раскомментируйте ее и замените `localhost` на `*`:

```
listen_addresses = '*' # Разрешить слушать на всех IP-адресах
```

??? 2: ????????? ??????? ????????????? (pg_hba.conf)

Это самый важный файл для управления доступом. Он определяет, **кто**, **откуда** и **как** может подключаться.

Откройте файл:

```
sudo nano /etc/postgresql/14/main/pg_hba.conf
```

Добавьте в **конец** файла новую строку, разрешающую подключение для вашего IP-адреса:

```
# TYPE  DATABASE        USER            ADDRESS                 METHOD
host    all             all             192.168.1.10/32         scram-sha-256
```

Разбор строки:

- `host`: Означает подключение по TCP/IP (через сеть).
- `all` (DATABASE): Правило применяется ко всем базам данных.
- `all` (USER): Правило применяется для всех пользователей.
- `192.168.1.10/32`: IP-адрес, с которого разрешено подключение. `/32` означает один конкретный IP. Можно указать подсеть, например, `192.168.1.0/24`.
- `scram-sha-256`: Метод аутентификации. Это современный и безопасный метод, требующий ввода пароля.

Пример из жизни: Вы — системный администратор, и ваш рабочий компьютер имеет статический IP `192.168.1.10`. Вы добавляете это правило, чтобы только вы (и никто другой из сети) могли подключаться к серверу баз данных с помощью pgAdmin для администрирования.

??? 3: ?????????? PostgreSQL

Чтобы все изменения вступили в силу, перезапустите службу.

```
sudo systemctl restart postgresql
```

Теперь вы можете открывать pgAdmin, нажимать "Add New Server" и вводить IP-адрес вашего сервера, имя пользователя и пароль для подключения.