????????? ? ??????????
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
```

Revision #1
Created 4 October 2025 23:11:45 by Admin
Updated 4 October 2025 23:12:43 by Admin