

postgres Basics

<https://computingforgeeks.com/how-to-install-postgresql-13-on-ubuntu/>

Shell Zugang

```
thommie@db2b:~$ sudo su postgres
```

```
postgres@db2b:/home/thommie$ psql
psql (13.6 (Ubuntu 13.6-1.pgdg20.04+1))
Type "help" for help.
postgres=#
```

\l listet alle DBs. Danach zu einer DB verbinden

```
postgres=# \c openproject postgres
You are now connected to database "openproject" as user "postgres".
```

\dt listet alle Tabellen in einer Datenbank auf

Schemata: <https://www.postgresqltutorial.com/postgresql-administration/postgresql-schema/> Das Standard Schema ist public

Alle Tabellen löschen

```
DO $$ DECLARE
  r RECORD;
BEGIN
  FOR r IN (SELECT tablename FROM pg_tables WHERE schemaname =
current_schema()) LOOP
    EXECUTE 'DROP TABLE ' || quote_ident(r.tablename) || ' CASCADE';
  END LOOP;
END $$;
```

\du listet alle User

User Passwort ändern

```
ALTER ROLE [username] WITH PASSWORD 'xxxx';
```

Zugriffsrechte auf Datenbank

```
GRANT ALL ON DATABASE [dbname] TO [username];

postgres=# GRANT ALL ON DATABASE keycloak TO keycloak;
GRANT
```

\s command history

\q DB shell beenden

Duplizieren einer Datenbank

Zuerst müssen alle laufenden Connections getrennt werden

```
SELECT pg_terminate_backend(pg_stat_activity.pid) FROM pg_stat_activity
WHERE pg_stat_activity.datname = 'Discourse' AND pid <> pg_backend_pid();
```

Danach wird kopiert

Dump der Datenbanken

1. Login auf Postgress Server
2. sudo nach postgres
3. Wechsel ins Home Dir von postgres (wg. Schreibrechten für sql File)

```
root@db2b:/home/thommie# sudo su postgres
postgres@db2b:/home/thommie$ cd ~
postgres@db2b:~$ pwd
/var/lib/postgresql
```

mit psql \l Namen aller DBs auflisten

```
postgres@db2b:~$ psql
psql (13.6 (Ubuntu 13.6-1.pgdg20.04+1))
Type "help" for help.
postgres=# \l
```

Danach die Datenbanken dumpen:

```
pg_dump pixelfed >> pixelfed_tokoeka.sql
```

Datenbank löschen

```
$ dropdb your_database
```

Verbindungstest zum postgresql Server

Test einer postgres Verbindung

pg_isready ist Teil von postgresql-client

```
pg_isready -d <db_name> -h <host_name> -p <port_number> -U <db_user>
```

Praktisch:

```
root@docker2:/var/discourse# pg_isready -d discourse -h 10.10.10.18 -p 5432  
-U discourse  
10.10.10.18:5432 - Verbindungen werden angenommen
```

Import der Datenbanken

```
psql -U postgres -W -d keycloak -f keycloak_tokoeka.sql
```

Alte Daten löschen

Datenbank

```
drop DATABASE onlyoffice;
```

User/Rolle

Auflisten der User

```
\du
```

Löschen eines Users

```
drop user IF EXISTS dendrite;
```

Datenbank duplizieren

Alle vorhandenen connections schliessen

```
SELECT pg_terminate_backend(pg_stat_activity.pid)  
FROM pg_stat_activity  
WHERE pg_stat_activity.datname = 'Discourse'  
AND pid <> pg_backend_pid();
```

Kopie

```
CREATE DATABASE Discoursdev
```

WITH TEMPLATE Discourse
OWNER postgres;

Postgres und UTF8

Standardmässig werden neue Datenbanken mit der Zeichenkodierung SQL_ASCII angelegt. Das passt meistens, aber nicht immer. Mit dieser Methode werden DBs mit UTF8 Kodierung angelegt:

<https://www.shubhamdipt.com/blog/how-to-change-postgresql-database-encoding-to-utf8/>

Für template1 das encoding auf UTF8 ändern

```
postgres=# UPDATE pg_database SET datistemplate = FALSE WHERE datname =  
'template1';  
postgres=# DROP DATABASE template1;  
postgres=# CREATE DATABASE template1 WITH TEMPLATE = template0 ENCODING =  
'UTF8';  
postgres=# UPDATE pg_database SET datistemplate = TRUE WHERE datname =  
'template1';  
postgres=# \c template1;  
You are now connected to database "template1" as user "postgres".  
template1=# VACUUM FREEZE;
```

Neu angelegte DBs auf Basis von template1 haben danach utf8 encoding

From:

<https://wiki.netzwissen.de/> - netzwissen.de Wiki

Permanent link:

<https://wiki.netzwissen.de/doku.php?id=postgres&rev=1728630180>

Last update: **11/10/2024 - 07:03**

