

# Wichtige Postgres Befehle

## Das Äquivalent zum MySQL "show databases" Befehl

```
# /usr/bin/sudo -u postgres psql -l
      List of databases
  Name      | Owner      | Encoding
-----+-----+-----
 app0000   | app0000   | UTF8
 app0001   | app0001   | UTF8
 app0002   | app0002   | UTF8
 app0003   | app0003   | UTF8
 app0004   | app0004   | UTF8
 ...
 postgres  | postgres  | UTF8
 template0 | postgres  | UTF8
 template1 | postgres  | UTF8
(29 rows)
```

oder auch

```
# /usr/bin/sudo -u postgres psql
Welcome to psql 8.1.18, the PostgreSQL interactive terminal.

Type: \copyright for distribution terms
      \h for help with SQL commands
      \? for help with psql commands
      \g or terminate with semicolon to execute query
      \q to quit

postgres=# \l
      List of databases
  Name      | Owner      | Encoding
-----+-----+-----
 app0000   | app0000   | UTF8
 app0001   | app0001   | UTF8
 app0002   | app0002   | UTF8
 app0003   | app0003   | UTF8
 app0004   | app0004   | UTF8
 ...
 postgres  | postgres  | UTF8
 template0 | postgres  | UTF8
 template1 | postgres  | UTF8
(29 rows)
```

## Das Äquivalent zum MySQL "show tables" Befehl

```
postgres=# \c app0000
You are now connected to database "app0000".
app0000=# \d
                List of relations
 Schema |          Name          | Type  | Owner
-----+-----+-----+-----
 public | attachmentdata        | table | app0000
 public | attachments            | table | app0000
 public | bandana                | table | app0000
 public | bodycontent            | table | app0000
 public | clustersafety         | table | app0000
 public | confancestors         | table | app0000
 ...
(40 rows)
```

Das Äquivalent zum MySQL "describe [table](#)" Befehl

```
postgres=# \c app0000
You are now connected to database "app0000".
```

```
app0000=# \d content
```

Column	Type	Modifiers
contentid	bigint	not null
contenttype	character varying(255)	not null
title	character varying(255)	
version	integer	
creator	character varying(255)	
creationdate	timestamp without time zone	
lastmodifier	character varying(255)	
lastmoddate	timestamp without time zone	
versioncomment	text	
prevver	bigint	
content_status	character varying(255)	
spaceid	bigint	
child_position	integer	
parentid	bigint	
messageid	character varying(255)	
draftpageid	character varying(255)	
draftspacekey	character varying(255)	
drafttype	character varying(255)	
draftpageversion	integer	
pageid	bigint	
parentcommentid	bigint	
username	character varying(255)	

Indexes:

```
"content_pkey" PRIMARY KEY, btree (contentid)
"c_contenttype_idx" btree (contenttype)
"c_draftpageid_idx" btree (draftpageid)
"c_drafttype_idx" btree (drafttype)
"c_messageid_idx" btree (messageid)
"c_pageid_idx" btree (pageid)
"c_parentcommid_idx" btree (parentcommentid)
"c_parentid_idx" btree (parentid)
"c_prevver_idx" btree (prevver)
"c_spaceid_idx" btree (spaceid)
"c_title_idx" btree (title)
"c_username_idx" btree (username)
```

Foreign-key constraints:

```
"fk6382c05917d4a070" FOREIGN KEY (prevver) REFERENCES content(contentid)
"fk6382c05974b18345" FOREIGN KEY (parentid) REFERENCES content(contentid)
"fk6382c0598c38fbea" FOREIGN KEY (pageid) REFERENCES content(contentid)
"fk6382c059b2dc6081" FOREIGN KEY (spaceid) REFERENCES spaces(spaceid)
"fk6382c059b97e9230" FOREIGN KEY (parentcommentid) REFERENCES content(contentid)
```

## Full Backup und Restore

```
$ pg_dump --user username --host hostname -b -F c dbname > dbname.dump
```

```
$ pg_restore --user username --host hostname -c -d dbname dbname.dump
```

```
$ sudo -u postgres pg_dumpall > full.sql
```

Eine einzelne DB kann mit dem Kommando `pg_dump` gesichert werden. Durch die Option `-F c` wird ein internes, komprimiertes Format gewählt, was bei einem späteren Import die meisten Möglichkeiten bietet.

Bei einem Import mit `pg_restore` gibt es unter Umständen (wenn man einen anderen User verwendet) zahlreiche Warnungen, die man aber ignorieren kann.