

PostgreSQL

auf Debian System

Inhaltsverzeichnis

- 1. Installation**
 - 1.1 Installation**
 - 1.2 Mit postgres anmelden**
- 2. Benutzer anlegen**
- 3. Datenbank anlegen**
 - 3.1 Optionen**
 - 3.2 Datenbank „mydatabase“ anlegen**
- 4. Datenbank auswählen**
 - 4.1 Datenbank auswählen**
 - 4.2 Datenbank abwählen**
- 5. Per TCP-IP auf Datenbank zugreifen**
 - 5.1 Konfiguration am Server**
 - 5.2 Konfiguration am Client**
 - 5.3 Verbindung herstellen**
- 6. Datenbank Funktionen**
 - 6.1 Tabelle erstellen**
 - 6.2 Tabelle anzeigen**
 - 6.3 Datensätze einfügen**
 - 6.4 Tabelleninhalt ausgeben**
 - 6.5 Zugriffsrechte anzeigen**
- 7. PostgreSQL und PHP**
 - 7.1 PHP Script**
- 8. Quellen**

von Pikart Sergej

1.Installation

1.1 Installation

Als su anmelden

```
# su
```

```
debian# apt-get install postgresql
```

mit OK bestätigen

1.2 Mit postgres anmelden

Jetzt kann man sich als Datenbank-Administrator auf dem System anmelden (erst als root) und die Datenbank administrieren. Wenn das Packet installiert ist legt das System einen Superuser „postgres“ automatisch an.

```
black:/home/serg# su - postgres
```

```
postgres@black:~$
```

2.Benutzer anlegen

Nach der Installation kann man die PostgreSQL Befehle anwenden.

Informationen bekommen Sie unter:

<http://www.postgresql.org/files/documentation/books/pghandbuch/html/>

Mit „createuser“(-P steht für Passwort) wird ein Benutzer angelegt, es wird nach einem Passwort abgefragt und einigen Rechten für den Benutzer(mit zweite Option ist Vorsicht geraten)

```
postgres@black:~$ createuser -P serg
```

Geben Sie das Passwort des neuen Benutzers ein:

Geben Sie es noch einmal ein:

Soll der neue Benutzer Datenbanken erzeugen dürfen? (j/n) j

Soll der neue Benutzer weitere neue Benutzer erzeugen dürfen? (j/n) j

```
CREATE USER
```

3.Datenbank anlegen

Zum Erzeugen von Datenbanken wird in der Console „createdb“ benutzt.

3.1 Optionen

-h, --hostgibt den Namen des Zielhosts an, auf dem die Datenbank erzeugt werden soll

-p, --portdies ist die Nummer des Ports, den der Server nach Verbindungsanforderungen abhört

-U, --username der Benutzername, unter dem die Verbindung aufgebaut wird

-W,

--password diese Option erzwingt eine Passwortabfrage

dbname der Name der Datenbank, die erzeugt wird

z.B

```
$ createdb -h myhost -p 1234 -U myname -W testdb
```

```
CREATE DATABASE
```

3.2 Datenbank „mydatabase“ anlegen

Anlegen der Datenbank „mydatabase“ mit „createdb“. Ohne Optionen wird eine Datenbank auf „localhost“ mit Ihrem Benutzernamen und Standardport 5432 erzeugt.

```
serg@black:~$ createdb mydatabase
CREATE DATABASE
serg@black:~$
```

4. Datenbank auswählen

4.1 Datenbank auswählen

Man kann jetzt die Datenbank mit dem Benutzer „serg“ verwenden

```
serg@black:~$ psql mydatabase
Willkommen bei psql 7.4.19, dem interaktiven PostgreSQL-Terminal.
```

Geben Sie ein: `\copyright` für Urheberrechtsinformationen
`\h` für Hilfe über SQL-Anweisungen
`\?` für Hilfe über interne Anweisungen
`\g` oder Semikolon, um eine Anfrage auszuführen
`\q` um zu beenden

```
mydatabase=# (hiermit wird geöffnete Datenbank angezeigt)
```

4.2 Datenbanken auflisten

Anzeigen aller Datenbanken mit „\l“

```
mydatabase=# \l
Liste der Datenbanken
  Name | Eigentümer | Kodierung
-----+-----+-----
mydatabase | serg      | UNICODE
rdf       | serg      | UNICODE
template0 | postgres | UNICODE
template1 | postgres | UNICODE
test      | serg      | UNICODE
(5 Zeilen)
```

```
mydatabase=#
```

4.2 Datenbank abwählen

Beenden der Verbindung zur Datenbank mit „\q“

```
mydatabase=# \q
serg@black:~$
```

4.3 Kommandos

Anzeigen aller Kommandos mit „\?“

```
mydatabase=# \?
```

5. Per TCP-IP auf Datenbank zugreifen

5.1 Konfiguration am Server

Um das zu ermöglichen müssen folgende Konfigurationsdateien geändert werden
gedit /etc/postgresql/7.4/main/pg_hba.conf
die Zeile an Ende einfügen

```
host all all 192.168.99.110 255.255.255.0 md5
```

gedit /etc/postgresql/7.4/main/postgresql.conf
Diesen Eintrag ändern

```
tcpip_socket=true
```

damit wird von dem Rechner mit IP 192.168.99.110 Zugriff erlaubt

PostgreSQL Server neustarten

```
#/etc/init.d/postgresql-7.4 restart
```

5.2 Per Konfiguration am Client

Am Client muss ein Packet nachinstalliert werden

```
#apt-get install postgresql-client-7.4
```

5.3 Per Konfiguration am Client

Verbindung aufbauen mit PostgreSQL Server

```
# psql --host 192.168.99.101 --dbname=mydatabase --username=serg
```

6.1 Tabelle erstellen

Hier wird eine Tabelle angelegt mit CREATE TABLE
CHECK prüft die Werte bevor die eingefügt werden

```
mydatabase=# CREATE TABLE schueler(  
mydatabase(# id SERIAL PRIMARY KEY,  
mydatabase(# vname VARCHAR(100) NOT NULL CHECK (length(trim(vname)) > 2),  
mydatabase(# nname VARCHAR(100) NOT NULL CHECK (length(trim(nname)) > 2),  
mydatabase(# klasse VARCHAR(100) NOT NULL CHECK (length(trim(klasse)) > 2),  
mydatabase(# gdatum DATE NOT NULL,  
mydatabase(# raucher BOOLEAN);  
HINWEIS: CREATE TABLE erstellt implizit eine Sequenz »schueler_id_seq« für die »serial«-Spalte  
»schueler.id«  
HINWEIS: CREATE TABLE / PRIMARY KEY erstellt implizit einen Index »schueler_pkey« für Tabelle  
»schueler«  
CREATE TABLE
```

6.2 Tabelle anzeigen

```
mydatabase=# \d
```

```
Liste der Relationen
```

```
Schema | Name | Typ | Eigentümer
```

```
-----+-----+-----+-----  
public | fahrer | Tabelle | serg
```

```
public | fahrer_id_seq | Sequenz | serg
```

```
public | fahrt | Tabelle | serg
```

```
(3 Zeilen)
```

6.3 Datensätze einfügen

```
mydatabase=# INSERT INTO schueler (
```

```
mydatabase(# vname, nname, klasse, gdatum, raucher)
```

```
mydatabase=# VALUES ('Anton', 'Kwashnin', 'IAV1', '05.06.2002', TRUE);
```

```
INSERT 17205 1
```

```
mydatabase=#
```

6.4 Tabelleninhalt ausgeben

```
mydatabase=# SELECT * FROM schueler;
```

```
id | vname | nname | klasse | gdatum | raucher
```

```
-----+-----+-----+-----+-----+-----  
1 | Anton | Kwashnin | IAV1 | 2002-05-06 | t
```

```
2 | Peter | Schwarz | IAV3 | 1991-05-06 | t
```

```
3 | Alexander | Schoenfeld | IAV4 | 1982-05-06 | f
```

```
(3 Zeilen)
```

```
mydatabase=#
```

6.5 Zugriffsrechte anzeigen

```
mydatabase=# \z
```

```
Zugriffsrechte für Datenbank »mydatabase«
```

```
Schema | Tabelle | Zugangsrechte
```

```
-----+-----+-----  
public | fahrer |
```

```
public | fahrer_id_seq |
```

```
public | fahrt |
```

```
public | schueler | {serg=a*r*w*d*R*x*t*/serg,apache2=arwdRxt/serg}
```

```
public | schueler_id_seq |
```

```
(5 Zeilen)
```

```
mydatabase=#
```

7.PostgreSQL und PHP

Man geht davon aus, dass apache mit allen zugehörigen Paketen installiert ist.
Pakete nachinstallieren für PostgreSQL:
`#apt-get install php5-pgsql`

7.1 PHP Script

```
<html>
<head><title>Tabelleninfo</title></head>
<body>
<?php

$host = "localhost";
$port = "5432";
$user = "serg";
$password = "rdf0608";
$dbname = "mydatabase";
echo "Test<br><br>";

$conn = pg_connect ("dbname=$dbname user=$user
    host=$host port=$port password=$password")
    or die ("Konnte keine Verbindung herstellen.");
if($conn){echo "Verbindung könnte hergestellt werden\n";}
$db = pg_dbname ($conn);
echo "Name der DB: '$db.'<br>";
$res = pg_query ("select * from schueler");
$num = pg_num_fields ($res);
echo "Anzahl der Felder: '$num.'<br>";
?>
<table border="1">
<tr align="left"><th>Feld</th><th>Name</th>
<th>Typ</th><th>Groesse</th></tr>
<?php
for ($i = 0; $i < $num; $i++) {
echo '<tr><td>'. $i. '</td>';
echo '<td>'.pg_field_name ($res, $i). '</td>';
echo '<td>'.pg_field_type ($res, $i). '</td>';
echo '<td>'.pg_field_size ($res, $i). '</td>';
}
echo '</table>';
?>
</body></html>
```

Mit dem Script werden die Informationen zur Tabelle ausgegeben

8.Quellen

<http://www.postgresql.org/files/documentation/books/pghandbuch/html/>
http://www.dynamicwebpages.de/50.tutorials.php?dwp_tutorialID=26