

22.16.0 Exkurs: PostgreSQL-Datenbankserver – Installation, Konfiguration und Test

Wenn Sie mit PostgreSQL-Datenbanken auf Ihrem System bearbeiten wollen, dann benötigen Sie den PostgreSQL-Datenbankserver.

Inhaltsverzeichnis

22.16.0 Exkurs: PostgreSQL-Datenbankserver – Installation, Konfiguration und Test.....	1
22.16.0.1 Hinweise.....	1
22.16.0.2 Installation des PostgreSQL-Datenbankservers.....	1
22.16.0.3 Passwort für den DB-Benutzer postgres festlegen.....	1
22.16.0.4 Status und Version.....	2
22.16.0.5 Dokumentation.....	2
22.16.0.6 Log-Dateien.....	3
22.16.0.7 Steuerung des PostgreSQL-Datenbankservers.....	3
22.16.0.8 Fernzugriff auf den PostgreSQL-Server.....	3
22.16.0.8.1 Installation PostgreSQL-Client.....	3
22.16.0.8.2 Konfiguration des PostgreSQL-Servers – Teil 1.....	3
22.16.0.8.3 Konfiguration des PostgreSQL-Servers – Teil 2.....	4
22.16.0.9 Übung.....	5
22.16.0.10 Option – Installation von Administrationsprogrammen.....	5
22.16.0.11 De-Installation des PostgreSQL-Datenbankservers.....	6

22.16.0.1 Hinweise

Die folgenden Hinweise und Anregungen sollten Sie beachten:

- In der vorliegenden Beschreibung werden die Installation, Konfiguration und ein Vorschlag für einen ersten Test des PostgreSQL-Datenbankservers für Linux Mint 20.1 beschrieben.
- Es wird die Version 12 des Datenbankserver PostgreSQL eingesetzt (Stand: 07.08.2022).
- Auf die Dokumentation zum PostgreSQL-Datenbankserver wird nur verwiesen.

22.16.0.2 Installation des PostgreSQL-Datenbankservers

So können Sie feststellen, welche PostgreSQL-Pakete bereits auf Ihrem System installiert sind:

```
$ dpkg --get-selections | grep postgres
ii gamba3-gb-db-postgresql 3.16.0+git20210419...~ubuntu20.04.1 amd64
PostgreSQL driver for the Gamba database
```

Der PostgreSQL-Datenbankserver kann über die Anwendungsverwaltung installiert werden. Oder Sie geben in einem Terminal folgende Zeilen ein, wobei das letzte Paket zusätzliche Features und Funktionalitäten hinzufügt (Option). Automatisch wird auch der PostgreSQL-Konsolen-Client *psql* installiert.

```
$ sudo apt-get update
$ sudo apt-get upgrade
$ sudo apt-get install postgresql postgresql-contrib
```

Die installierte Version des PostgreSQL-Datenbankservers zeigen Sie so an:

```
$ pg_config | grep 'VERSION'
VERSION = PostgreSQL 12.6 (Ubuntu 12.6-0ubuntu0.20.04.1)
```

Anschließend sollten Sie auch die aktuelle Dokumentation (hier für die Version 12) des Datenbankserver installieren:

```
$ sudo apt-get install postgresql-doc postgresql-doc-12
```

22.16.0.3 Passwort für den DB-Benutzer postgres festlegen

Während der Installation des PostgreSQL-Datenbankservers wird automatisch der Datenbankbenutzer `postgres` als Super-User angelegt.

Standardmäßig ist nach der Installation die Passwort-Authentifizierung für den PostgreSQL-Administrator `postgres` deaktiviert. Der einfachste Weg, um das Konto für den DB-Benutzer `postgres` mit einem Passwort zu vervollständigen, führt über das PostgreSQL-Client-Programm 'psql':

```
hans@mint20:~$ sudo -u postgres psql postgres
psql (12.6 (Ubuntu 12.6-0ubuntu0.20.04.1))
Type "help" for help.

postgres=# help
You are using psql, the command-line interface to PostgreSQL.
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=# \password postgres
Enter new password:          → Das hier frei gewählte Passwort 'appw' wird verdeckt und ohne Echo eingegeben
Enter it again:              → Das gewählte Passwort wiederholen
postgres=# \q
hans@mint20:~$
```

Jetzt können Sie das Client-Programm 'psql' mit dem Konto des Administrators (DB-Benutzer: postgres, DB-Benutzer-Passwort: appw) auf 127.0.0.1 oder localhost auf dem Port 5432 starten und sich mit dem Befehl \dg oder \dg+ die Eigenschaften des DB-Administrators postgres anzeigen lassen:

```
hans@mint20:~$ psql -h localhost -p 5432 -U postgres -W
Password:
psql (12.6 (Ubuntu 12.6-0ubuntu0.20.04.1))
SSL connection (protocol:TLSv1.3, cipher: TLS_AES_256_GCM_SHA384, bits: 256, compression: off)
Type "help" for help.
postgres=# \dg

```

Role name	List of roles Attributes	Member of
postgres	Superuser, Create role, Create DB, Replication, Bypass RLS	{}

```
postgres=# \q
hans@mint20:~$
```

Achtung: Verwenden Sie den Datenbankbenutzer `postgres` ausschließlich für die Administration des PostgreSQL-Datenbankservers!

22.16.0.4 Status und Version

So ermitteln Sie den Status des PostgreSQL-Datenbankservers, da der PostgreSQL-Datenbankserver nach der Installation sofort gestartet wird:

```
$ service postgresql status
```

Achtung! Der PostgreSQL-Datenbankserver wird bei jedem Systemstart *automatisch* gestartet (Standard). Das können Sie mit diesen beiden Befehlen ändern:

```
$ sudo systemctl disable postgresql -- PostgreSQL aus der Autostart-Liste entfernen
$ sudo systemctl enable postgresql -- PostgreSQL der Autostart-Liste wieder hinzufügen
```

Wenn der PostgreSQL-Datenbankserver beim Systemstart nicht gestartet wird, dann können Sie ihn zum Beispiel mit 'sudo service postgresql start' selbst starten.

So ermitteln Sie alternativ die aktuelle Version des PostgreSQL-Datenbankservers:

```
$ psql -V
psql (PostgreSQL) 12.11 (Ubuntu 12.11-0ubuntu0.20.04.1)
```

22.16.0.5 Dokumentation

Eine umfangreiche Dokumentation zum PostgreSQL-Datenbankserver finden Sie unter:

```
LINK1: https://www.postgresql.org/docs/12/index.html
```

Hilfe gibt es auch *lokal* für die Version 12:

LINK2: <file:///usr/share/doc/postgresql-doc-12/html/index.html>

22.16.0.6 Log-Dateien

Die letzten Einträge der *aktuellen* Log-Datei des PostgreSQL-Datenbankservers können Sie sich mit dem folgenden Befehl in einer Konsole ansehen:

```
$ sudo tail -f /var/log/postgresql/postgresql-12-main.log
```

22.16.0.7 Steuerung des PostgreSQL-Datenbankservers

Sie können einen PostgreSQL-Datenbankserver starten, stoppen, neu-starten, die Konfigurationsdatei neu einlesen und den Status abfragen. Die folgenden Aufrufe in einem Terminal steuern den PostgreSQL-Datenbankserver mit den in der Liste aufgeführten Parametern:

```
$ sudo /etc/init.d/postgresql {start|stop|restart|reload|force-reload|status}
```

Alternativ:

```
$ sudo service postgresql {start|stop|restart|reload|force-reload|status}
```

Beispiele:

```
$ /etc/init.d/postgresql status          → Nur die Status-Abfrage erfordert keine erhöhten Rechte
$ systemctl status postgresql.service
```

```
$ sudo /etc/init.d/postgresql reload
[sudo] Passwort für hans:
Reloading postgresql configuration (via systemctl): postgresql.service.
```

22.16.0.8 Fernzugriff auf den PostgreSQL-Server

Wenn Sie von entfernten Hosts im Netzwerk auf den PostgreSQL-Server zugreifen wollen, dann sind die folgenden Voraussetzungen zu erfüllen:

- Auf den entfernten Computern ist das Paket postgresql-client installiert und Sie können den PostgreSQL-Client `psql` nutzen, wenn das notwendig sein sollte.
- Der PostgreSQL-Server ist in der Konfigurationsdatei `/etc/postgresql/12/main/postgresql.conf` für Verbindungsanforderungen im (lokalen) Netz konfiguriert.
- Zusätzlich ist in der Konfigurationsdatei `/etc/postgresql/12/main/pg_hba.conf` festgelegt, welche Hosts (Hostnamen) aus welchen Netzen (IP-Ranges) Zugriff auf ausgewählte PostgreSQL-Datenbanken erhalten. Der Suffix 'hba' steht für 'host based access'.

22.16.0.8.1 Installation PostgreSQL-Client

Der (Konsolen-)Client `psql` ist schnell installiert:

```
$ sudo apt update
$ sudo apt install postgresql-client
```

22.16.0.8.2 Konfiguration des PostgreSQL-Servers – Teil 1

Mit dem folgenden Befehl können Sie prüfen, an welchen IP-Adressen der PostgreSQL-Server aktuell lauscht:

```
$ ss -nlt
State      Recv-Q   Send-Q   Local Address:Port   Peer Address:Port   Process
...
LISTEN    0         244      127.0.0.1:5432       0.0.0.0:*
...
$
```

Das ist korrekt, denn standardmäßig kann nur ein PostgreSQL-Client, der auf demselben Computer wie der PostgreSQL-Server gestartet wurde, eine Verbindung zu ihm lokal herstellen.

Ändern Sie mit erhöhten Rechten in der Konfigurationsdatei `/etc/postgresql/12/main/postgresql.conf`

```
$ sudo xed /etc/postgresql/12/main/postgresql.conf
```

nur den originalen Eintrag von

```
listen_addresses = 'localhost'
```

auf

```
listen_addresses = '*'
```

um sicherzustellen, dass der PostgreSQL-Server nun auf allen IP-Adressen – in Abhängigkeit von den Eintragungen in der Konfigurationsdatei `/etc/postgresql/12/main/pg_hba.conf` – lauscht. Nach dem Speichern der neuen Konfiguration schließen Sie den Editor und starten Sie den PostgreSQL-Server mit

```
$ sudo systemctl restart postgresql
```

Jetzt können Sie erneut prüfen, auf welchen Netzwerkschnittstellen der PostgreSQL-Server lauscht:

```
$ ss -nlt
State      Recv-Q   Send-Q   Local Address:Port   Peer Address:Port   Process
LISTEN     0         244      0.0.0.0:5432          0.0.0.0:*
LISTEN     0         244      [::]:5432           [::]:*
...
hans@pc-a-mint20:~$
```

22.16.0.8.3 Konfiguration des PostgreSQL-Servers – Teil 2

Ändern Sie die Konfigurationsdatei `"pg_hba.conf"` so, dass Remote-Clients eine Verbindung zu dem Datenbank-Server herstellen können.

- Sichern Sie das Original der Konfigurationsdatei `/etc/postgresql/12/main/pg_hba.conf`.
- Stoppen Sie den PostgreSQL-Server, wenn er bereits gestartet wurde.
- Öffnen Sie die Datei `"pg_hba.conf"` in einem Texteditor Ihrer Wahl.
- Fügen die vorgesehenen Zugriffsrechte wie folgt hinzu: Datenbanken, DB-Benutzer, IP-Bereiche, Authentifizierungsmethode.

Sichern Sie das Original der Konfigurationsdatei:

```
$ sudo cp /etc/postgresql/12/main/pg_hba.conf /etc/postgresql/12/main/pg_hba.conf.old
$ sudo /etc/init.d/postgresql stop
$ sudo xed /etc/postgresql/12/main/pg_hba.conf
```

Vorschlag für eine Testumgebung:

In der Konfigurationsdatei `/etc/postgresql/12/main/pg_hba.conf` fügen Sie die folgenden beiden Zeilen oberhalb der anderen Einträge ein, um von allen Hosts aus für alle DB-User (mit einem gesetzten Passwort) den Zugriff zu allen Datenbanken zu ermöglichen:

```
# TYPE DATABASE USER  CIDR-ADDRESS METHOD
host all all  0.0.0.0/0 md5
```

Sie können aber auch die Zugriffsrechte restriktiver vergeben:

```
# TYPE DATABASE USER  CIDR-ADDRESS METHOD
host test all  192.168.100.0/24 md5
```

Jetzt gilt einschränkend: Alle DB-Benutzer dürfen von allen Remote-Clients, deren IP-Adresse mit `192.168.100` beginnt, auf die Datenbank `'test'` zugreifen.

Speichern und schließen Sie die Konfigurationsdatei `/etc/postgresql/12/main/pg_hba.conf`. Starten Sie den PostgreSQL-Server wieder:

```
$ sudo /etc/init.d/postgresql restart
```

22.16.0.9 Übung

Die Übung – ausgeführt in einer Konsole mit dem PostgreSQL-Monitor `psql` (Client) – besteht darin, dass Sie wichtige Befehle noch einmal wiederholen und dabei

- zuerst den Status des PostgreSQL-Datenbankservers ermitteln,
- dann den Server (auf dem Standard-Port 5432 starten), wenn das notwendig ist,
- danach den PostgreSQL-Client *psql* mit dem Konto (*postgres*, *appw*) starten,
- anschließend die Hilfe mit *help* aufrufen und darin zum Beispiel abfragen, wie eine neue Datenbank angelegt wird und
- abschließend den PostgreSQL-Client mit der Anweisung *\q* oder **CTRL+D** beenden. Ab der PostgreSQL-Version 11 können Sie auch *exit* oder *quit* verwenden.

```
$ service postgresql status
$ sudo service postgresql start
$ psql -h localhost -p 5432 -U postgres -W
postgres=# help
postgres-# \h CREATE DATABASE
postgres-# \q
$
```

22.16.0.10 Option – Installation von Administrationsprogrammen

Ein Programm zur Administration von *PostgreSQL-Datenbanken* ist *pgAdmin3*, dessen graphische Benutzeroberfläche u.a. die Administration und Entwicklung von Datenbanken, DB-Tabellen sowie das Anlegen von DB-Benutzern erleichtert.

Das Programm *pgAdmin3* können Sie über die Anwendungsverwaltung installieren. Alternativ geben Sie in einem Terminal folgende Zeile ein:

```
$ sudo apt install pgadmin3
```

Beachten Sie jedoch, dass *pgAdmin3* derzeit (Stand: 8. November 2021) als veraltet gilt und bei einer neuen Version von PostgreSQL (Version ≥ 12) nicht mehr korrekt funktioniert!

Auf der Seite <https://technium.ch/ubuntu-2104-pgadmin4-installieren-tutorial/> finden Sie eine Anleitung, wie Sie das Programm *pgAdmin4* installieren. Als sehr guten Ersatz für *pgAdmin3/4* können Sie das Programm *DBeaver* über ein PPA installieren und automatisch aktuell halten:

```
$ sudo add-apt-repository ppa:serge-rider/dbeaver-ce
$ sudo apt-get update
$ sudo apt-get install dbeaver-ce
```

Nach der Installation können Sie neben Ihren PostgreSQL-Datenbanken auch die Datenbanken, Datenbanktabellen und Datenbank-Benutzer für weitere bekannte Datenbank-Typen wie SQLite oder MySQL/MariaDB schnell und zuverlässig in *einem* Programm verwalten:



Abbildung 22.16.0.10.1: Verbindungen zu vielen DBMS verwalten

Vor allem bei Datenbanken mit mehreren DB-Tabellen werden Sie die grafischen Darstellungen der bestehenden Relationen des Programms DBeaver zu schätzen wissen:

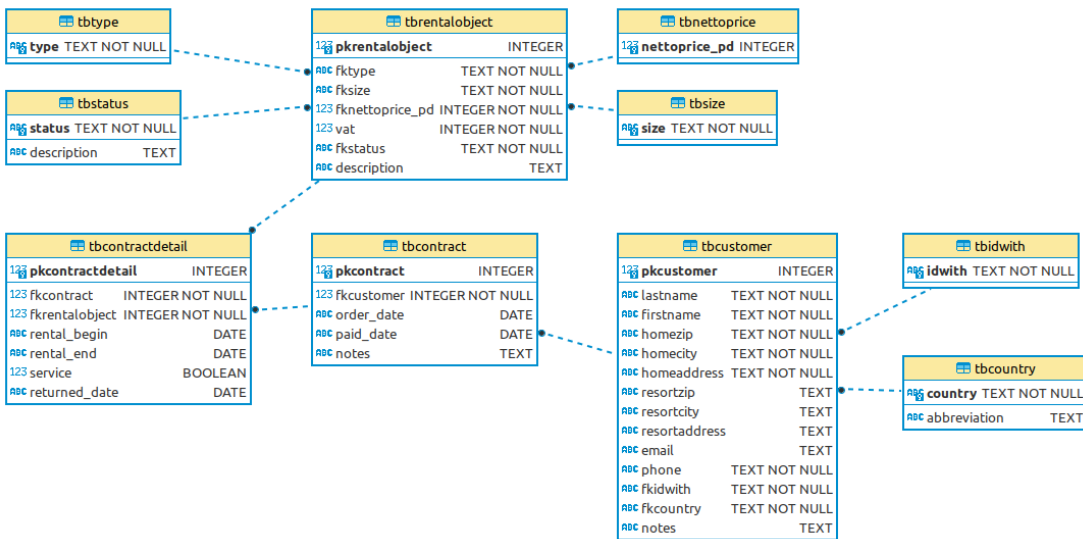


Abbildung 22.16.0.10.2: ER-Diagramm einer SQLite-Datenbank (Entitätsbeziehungen)

22.16.0.11 De-Installation des PostgreSQL-Datenbankservers

Für den Fall einer De-Installation des Datenbank-Servers ist es immer eine gute Idee, wenn Sie von den bisher verwendeten Datenbanken eine komplette Sicherung von Struktur und Daten anlegen oder nur von der Struktur (Schema) ausgewählter Datenbanken:

```
$ pg_dump --host=localhost --port=5432 --username=test --inserts --dbname=test --file=test.sql
$ pg_dump --host=localhost --port=5432 --username=test --schema-only --dbname=test --file=test.schema.sql
```

Mit dem Befehl dpkg für die Verwaltung von Debian-Paketen können Sie nach allen installierten Paketnamen suchen, die den Teil-String 'postgres' enthalten.

```
hans@pc-a-mint20:~$ dpkg -l | grep postgres
ii gamba3-gb-db-postgresql 3.17.3+git202..ubuntu20.04.1 amd64 PostgreSQL driver for the Gamba database
ii odbc-postgresql:amd64 1:12.01.0000-1 amd64 ODBC driver for PostgreSQL
ii postgresql 12+214ubuntu0.1 all object-relational SQL database (supported version)
ii postgresql-12 12.11-0ubuntu0.20.04.1 amd64 object-relational SQL database, version 12 server
ii postgresql-client-12 12.11-0ubuntu0.20.04.1 amd64 front-end programs for PostgreSQL 12
ii postgresql-client-common 214ubuntu0.1 all manager for multiple PostgreSQL client versions
ii postgresql-common 214ubuntu0.1 all PostgreSQL database-cluster manager
ii postgresql-contrib 12+214ubuntu0.1 all additional facilities for PostgreSQL (supported version)
ii postgresql-doc 12+214ubuntu0.1 all documentation for the PostgreSQL database management system
ii postgresql-doc-12 12.11-0ubuntu0.20.04.1 all documentation for the PostgreSQL database management system
hans@pc-a-mint20:~$
```

Wenn Sie sich von PostgreSQL – hier in der Version 12 – trennen wollen, dann de-installieren Sie entweder über die Synaptic-Paketverwaltung oder in einer Konsole mit folgenden Befehlen:

```
sudo apt-get --purge remove {POSTGRES-PACKAGE-NAME}
-----
sudo apt-get --purge remove postgresql
sudo apt-get --purge remove postgresql-12
sudo apt-get --purge remove postgresql-client-12
...
sudo apt-get --purge remove postgresql-doc-12
```

Abschließend können Sie die folgenden (Daten-)Verzeichnisse löschen:

```
$ sudo rm -rf /var/lib/postgresql/
$ sudo rm -rf /var/log/postgresql/
$ sudo rm -rf /etc/postgresql/
```