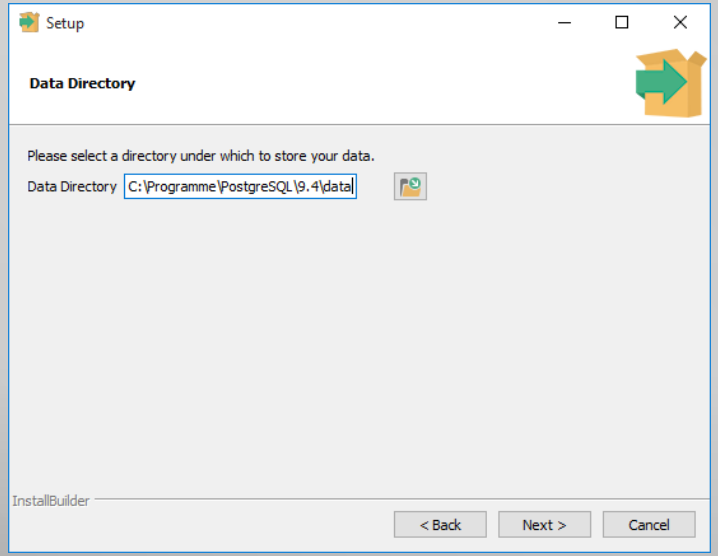
PostgreSQL: Tipps und Tricks

Installation

Aus dem Grund, dass nach Microsoft-Richtlinien keine Daten unter c:\Programme sich befinden sollte, ist ein anderes Installationsverzeichnis empfehlenswert, z.B. **C:\PostgreSQL**.

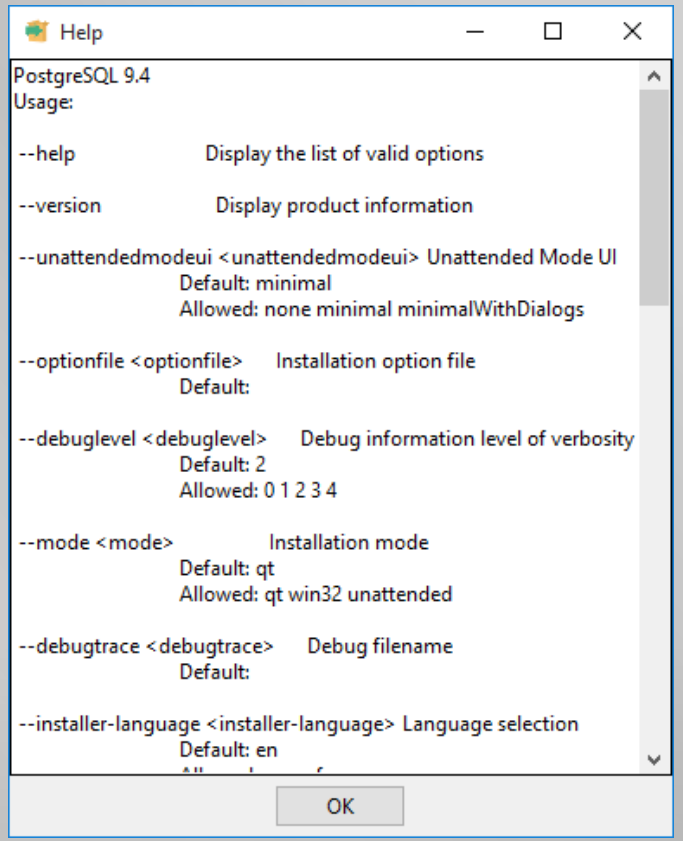


Der Windows-Benutzername, unter dem PostgreSQL installier wird, darf keine Leerzeichen im Anmeldenamen enthalten.

Unbeaufsichtigte Installation

Die Parameter dafür können mit der Hilfe des folgenden Befehls aufgerufen werden:

postgresql-9.4.4-3-windows-x64.exe --help



Unbeaufsichtigte Installation

Parameter	Notiz
--mode	Installationsmodus
--prefix	Installationsverzeichnis
--datadir	Datenverzeichnis
--serverport	Netzwerport
--superaccount	Superuser. Default: postgres
--superpassword	Superuser-Kennwort. Sollte vorgegeben werden, sonst kommt ein Dialog.
--servicename	Name für den Dienst. Leerzeichen im Namen können zu Problemen bei der Installation führen.
--serviceaccount	Benutzer für den Dienststart. Default: postgres
--servicepassword	Kennwort für den Dienststart. Sollte vorgegeben werden, sonst kommt ein Dialog.
--locale	Hier kann die Sprache so eingetragen werden, wie diese im Auswahldialog bei der normalen Installation steht.

postgresql-9.4.4-3-windows-x64.exe“ --mode unattended --prefix “C:\PostgreSQL\9.4” --datadir “C:\PostgreSQL\9.4\data” --locale "German, Germany" --serverport 5432 --servicename "postgresql-9" --serviceaccount “postgres" --servicepassword “postgres" --superaccount “postgres“ –superpassword “postgres”

Unbeaufsichtigte Deinstallation

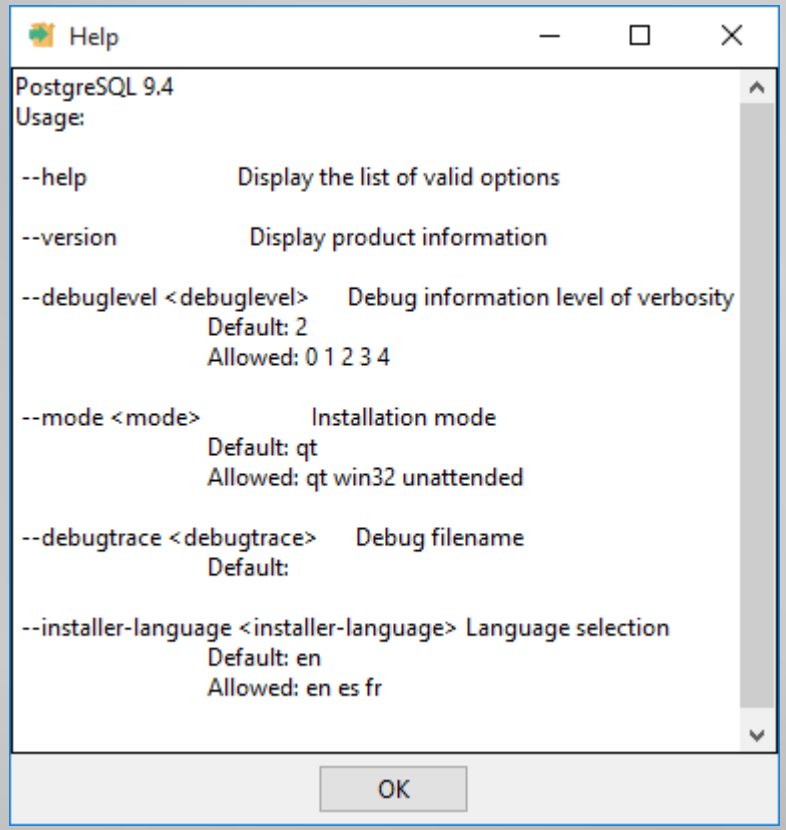
Die Parameter dafür können mit der Hilfe des folgenden Befehls aufgerufen werden:

uninstall-postgresql.exe --help

Diese Datei ist im Installationsverzeichnis zu finden, z.B. unter C:\PostgreSQL\9.4.

Tipp: Falls bei der Installation ein Fehler bei der Erstellung der Deinstallationsdatei auftritt, liegt es meistens an den Zugriffsrechten auf das Installationsverzeichnis.

C:\PostgreSQL\9.1\uninstall-postgresql.exe --mode unattended



Upgrade älterer Version

Wichtig ist, dass bei einer Änderung der Hauptversion (z.B. 9.1.4 auf 9.2.0) darf das Upgrade nicht eingespielt werden. Bei der Hauptversionsänderung werden immer die Systemtabellen geändert.

Vorgehen:

- Datenbanken sichern (pg_dumpall)
- Alten Server deinstallieren
- Neuen Server installieren
- Datenrücksicherung

Ein Update z.B. von 9.1.4 auf 9.1.5 ist normal ohne der oberen Schritte möglich.

Downgrade zwieschen der Hauptversionen ist nicht möglich.

ODBC-Treiber

Die Parameter für die unbeaufsichtigte De-/Installation der PostgreSQL-ODBC-Trieber können mit der Hilfe des folgenden Befehls aufgerufen werden:

Psqlodbc.msi –help

Beispiele:

für unbeaufsichtigte Installation ohne Benutzeroberfläche für alle Benutzer:
msiexec /qn /jm psqlodbc.msi

für eine Reparaturinstallation mit dem Erzwingen der Neuinstallation aller Dateien:
msiexec /fa psqlodbc.msi

und eine Deinstallation ohne Benutzeroberfläche:
msiexec /qn /x psqlodbc.msi

Einrichtung

Nach der Installation ist der PostgreSQL-Server evtl. aus dem Netzwerk nicht erreichbar. Dafür müssen 2 Schritte durchgeführt werden:

- Netzwerk Port (Standard 5432) in der Firewall freischalten
- In der Konfigurationsdatei „pg_hba.conf“ muss das Netzwerkzugriff freigeschaltet werden.

```
# TYPE  DATABASE        USER            ADDRESS                 METHOD

# IPv4 local connections:
host    all             all             127.0.0.1/32            md5
host    all             all             0.0.0.0/0               md5
# IPv6 local connections:
host    all             all             ::1/128                 md5
# Allow replication connections from localhost, by a user with the
# replication privilege.
#host    replication     postgres        127.0.0.1/32            md5
#host    replication     postgres        ::1/128                 md5
```

Speichereinstellungen

Nach der Installation von PostgreSQL sind die Speichereinstellungen des Servers so eingestellt, dass diese für die Entwicklung aber nicht für den normalen Betrieb ausreichend sind und müssen angepasst werden:

Einstellung	Beschreibung	Wirkung
Shared_buffer	10-25% des Arbeitsspeichers, mind. 128 kB oder 16*max_connections. Standard - ca. 32 MB je 1 GB Speicher. Gilt als Obergrenze und muss unbedingt angepasst werden.	Gesamter Datenbank-cluster und alle Datenbanken
Temp_buffers	Standard – 8MB. Gilt als Obergrenze, mind. 800 kB. Ändern nur, wenn mit temporären Tabellen gearbeitet wird.	Je Datenbank-verbindung
Work_mem	Standard - ca. 1 MB je 1 GB Speicher, mind. 64 kB. Sinvolle Änderung nur bei größeren Auswertungen (Sortierung, Filter, OLAP, Data Mining). Notwendig für Create Index, Alter Table, Vacuum, Cluster. Gilt als Obergrenze.	Pro Operation und kann von einer Abfrage auch gleichzeitig mehrmals verwendet werden.
Maintenance_work_mem	Standard - ca. 16 MB je 1 GB Speicher, mind. 1 MB. Notwendig für Create Index, Alter Table, Vacuum, Cluster. Gilt als Obergrenze.	Je Datenbanksitzung.

Speichereinstellungen

Die Speichereinstellungen können zum Teil in einer Sitzung verändert werden. Die Veränderung ist nur für diese Sitzung gültig.

Einstellung	Befehl
Shared_buffer	Kann nur vor dem Start der PostgreSQL-Instanz in der Konfiguration geändert werden!
Temp_buffers	Set temp_buffers TO `96MB`;
Work_mem	Set work_mem TO `32MB`;
Maintenance_work_mem	Set maintenance_work_memTO `512MB`;

Beispiel:

Set maintenance_work_mem TO `512MB`;
CREATE INDEX ….
RESET maintenance_work_mem;

Datensicherung

1. RAID
2. Replikation
3. Herkömmliche Dateisystemsicherung.
    a. Vor der Datensicherung wird es empfohlen, den PostgreSQL-Server zu stoppen und nach der Datensicherung wieder zu starten.
    b. Es reicht den Data-Verzeichnis zu sichern.
4. Dump. Geeignet bis 50 GB Datenumfang.
    a. Mitgelieferte PostgreSQL-Tools, die auch während der normalen Datenbankbenutzung verwendet und auch von anderen PCs ausgeführt werden können. Diese Methode kann auch in eigene Anwendungen integriert werden.
    b. Nachteil: keine inkrementelle Sicherung.
5. WAL-Archivierung (Write-Ahead-Log) – ermöglicht inkrementelle Sicherung und Hot Standby, ist aber etwas komplizierter in der Benutzung.

Datensicherung

Dump aller globalen Objekte (Benutzerrollen, Tablespaces und Definition der Datenbanken)

```
pg_dumpall.exe --host localhost --port 5432 --username "postgres" --database "postgres"  --verbose --file "C:\temp\postgres_globals.sql"
--globals-only
```

Dump aller Datenbanken und der globalen Objekte

```
pg_dumpall.exe --host localhost --port 5432 --username "postgres" --verbose --file "C:\temp\postgres_server.sql„
```

Dump einer einzelner Datenbank

```
pg_dump.exe --host localhost --port 5432 --username "postgres" --format custom --blobs --verbose --file
"C:\temp\postgres_test_db.backup" "test_db"
```

Hilfe aufrufen

```
pg_dumpall.exe -?
pg_dump.exe -?
```

Datenwiederherstellung

1. Bei Dateisystembasierter Datensicherung kann das Data-Verzeichnis einfach durch das gesicherte ersetzt werden, wobei der Server vorher gestoppt werden muss.
2. Dump
   a. Psql.exe für die Wiederherstellung der „sql“-Dateien.
   b. Pg_restore.exe für Custom-Format-Dateien.

```
psql.exe --host localhost --port 5432 --username "postgres" --file "C:\temp\postgres_test_db.sql" "test„

pg_restore.exe --host localhost --port 5432 --username "postgres" --dbname "test"  --verbose "C:\temp\postgres_test_db.backup"
```

Wartung

Befehl	Beschreibung
Vacuum [tabelle]	Gibt den Platz gelöschter Daten zur Wiederverwendung frei aber nicht den physischen Platz, kann aber währen des Produktivbetriebes ausgeführt werden. Bevorzugte Benutzung.
Vacuum full [tabelle]	Gibt auch den physischen Festplattenplatz frei, benötigt aber exklusiven Zugriff auf die Datenbank bzw. Tabelle.
Vacuum [full] freeze	Muss spätestens alle 2^{30} Transaktionen durchgeführt werden.
Analyse [tabelle]	Wartung der Planerstatistiken
Reindex { index | table | database | system } name	Indexe neu organisieren. Erfordert exklusiven Zugriff.

Autovacuum-Dienst ist standardmäßig aktiviert, kann aber zu ungünstigen Zeiten ausgeführt werden und System auslasten. Der Autovacuum kann aber konfiguriert werden.

Einbinden ins eigene Programm

Über die Shell-Ausführung aus eigenem Programm kann die Datensicherung/Wiederherstellung von einem beliebigen PC durchgeführt werden. Die Version der PostgreSQL-Dateien, die mit eigenem Programm ausgeliefert wird, sollte zu der Version des Servers passen.

Benötigte Dateien:

- Pg_dump.exe
- Pg_dumpall.exe
- Pg_restore.exe
- Psql.exe
- libintl-8.dll
- libpq.dll

Shell-Einstellung	Beschreibung
Set PGHOST=localhost	Server festlegen (Name, IP)
Set PGPORT=5432	Port festlegen
Set PGDATABASE=test	Datenbank festlegen
Set PGUSER=postgres	Benutzernamen festlegen
Set PGPASSWORD=postgres	Benutzerkennwort festlegen

Datenreparatur

- Festplatte defekt
  - Wenn die Festplatte defekt ist, aber das Data-Verzeichnis gerettet werden konnte, kann dieser bei einem neuinstallierten Server wieder eingespielt werden.
- Server startet nicht – Checkpoint-Fehler in Log
  - WAL (Write-Ahead-Log) ist defekt. Hier hilft der Befehl pg_resetxlog.exe, z.B. *pg_resetxlog.exe C:\PostgreSQL\9.4\data* und danach pg_resetxlog.exe –f C:\PostgreSQL\9.4\data
    Danach die Datenbank sichern, löschen und wiederherstellen.
    Bitte zur genaueren Info zum Befehl pg_restorxlog die Hilfe nachlesen!
- Datei im Datenbankverzeichnis (Struktur) fehlt. Sichern der Daten und Wartung sind nicht mehr möglich.
  - Die fehlende Datei durch eine Leere ersetzen und die Datensicherung wieder versuchen. In diesem Fall ist der Inhalt der fehlenden Datei endgültig verlören.
- PostgreSQL-Verzeichnis und -Netzwerkport als Ausnahme in der Antivieren-Software eintragen.

Export-/Import der Daten

Ob für einen Datenexport/-import oder für die Datenrettung, sind die folgenden Befehle interessant.

Export

copy (select kunden_nr, nachname, vorname, geburtstag, geschlecht, strasse, plz, ort from kunden where kunden_nr>1000) to 'c:\kunden.csv' with (format csv, delimiter ';', header 1, quote '"')

copy kunden to 'c:\Temp\kunden.csv' WITH (format csv, delimiter ';', header 1, quote '"')

Import

copy kunden (kunden_nr, nachname, vorname, geburtstag, geschlecht, strasse, plz, ort) from 'c:\Temp\kunden.csv' WITH (format csv, delimiter ';', header 1, quote '"')

copy kunden from 'c:\Temp\kunden.csv' WITH (format csv, delimiter ';', header 1, quote '"')

Geschwindigkeit der SQL-Befehle

EXPLAIN Select …

Der Befehl gibt den Ausführungsplan des SQL-Befehls aus. Bei großen Sequentiellen Suchwerten sollte über die zus. Indexe nachgedacht werden.

Mit dem folgenden Befehl wird der angegebene SQL-Befehl zus. noch wirklich ausgeführt, um die realistische Werte zu ermitteln (VORSICHT bei Update, Insert, Delete):

EXPLAIN ANALYSE Select …

Tabellen mit Beziehung

Tabellen mit Beziehung (Fremdschlüssel, Foreign Key) anlegen, so dass beim Löschen eines Datensatzes in der Haupttabelle auch die abhängigen Datensätze in der 2. Tabelle automatisch gelöscht werden.

CREATE TABLE kunden(kunden_id bigserial NOT NULL, vorname character(50), name character(50), CONSTRAINT kunden_id_ord PRIMARY KEY (kunden_id));

CREATE TABLE telefon(telefon_id serial NOT NULL, kunden_id bigint NOT NULL, typ character(20), nummer character(50), CONSTRAINT telefon_id_idx PRIMARY KEY (telefon_id), CONSTRAINT kunden FOREIGN KEY (kunden_id) REFERENCES kunden (kunden_id) MATCH SIMPLE ON UPDATE RESTRICT ON DELETE CASCADE)

Die meist verwendeten Parameter:

Parameter	Beschreibung
RESTRICT	Änderung verbieten
CASCADE	Änderung auch in der anderen Tabelle durchführen

Benutzerverbindungen

Manchmal ist es nötig die aktuellen Benutzerverbindungen zu ermitteln:

```
SELECT username, pid FROM pg_stat_activity WHERE datname = current_database();
```

Mit dem folgenden Befehl (ab PostgreSQL 8.4) können die Verbindungen auch beendet werden:

```
SELECT pg_terminate_backend(your_pid);
```

Nützliche SQL-Befehle

Befehl	Notiz
Select Version() oder *select name, setting from pg_settings where name in ('server_version', 'server_version_num')*	Die Serverversion abfragen
SELECT pg_stat_get_backend_pid(s.backendid) AS procpid, pg_stat_get_backend_activity(s.backendid) AS current_query FROM (SELECT pg_stat_get_backend_idset() AS backendid) AS s	Was macht gerade der Server (welche Befehle werden gerade ausgeführt)
select pg_database_size('test_db')	Datenbankgröße in Bytes
select pg_size_pretty(pg_database_size('test_db'))	Datenbankgröße in z.B. MB

Fragen?

Kontakt

Andreas Engler
Internet: https://www.engler-home.de

Download

http://www.postgresql.org/download

Quellen

„Versionskontrolle mit Git“ ISBN: 978-3-89721-945-8
http://www.postgresql.org/
https://de.wikibooks.org/wiki/Einf%C3%BChrung_in_SQL:_Fremdschl%C3%BCssel-Beziehungen