

Die ersten Schritte in der Administration der PostgreSQL-Datenbank



von Stefan Kunick

Inhaltsverzeichnis

Vorwort.....	5
Über die Datenbank PostgreSQL.....	6
Installation der Datenbank auf der LINUX-Host.....	7
Installation einer neueren Version aus dem Internet.....	8
Installation auf einer Host mit SuSE-LINUX.....	10
Installation mit der SuSE-Distribution.....	10
Der User postgres (SuSE).....	13
Installation auf einer Host mit Red Hat/Fedora-LINUX.....	15
Installation mit der Red Hat/Fedora-Distribution.....	15
Der User postgres (Red Hat/Fedora).....	17
Weiterführende Arbeiten.....	20
Automatisches Login.....	27
Die Datenbank beenden.....	28
Die Konfigurationsdatei pg_hba.conf.....	30
Die Konfigurationsdatei postgres.conf.....	33
Die Konfigurationsdatei pg_ident.conf.....	34
Das Programm postmaster.....	35
Die Parameter zum Programm postmaster.....	36
Arbeiten mit der Datenbank.....	38
Datenbanken anlegen.....	38
Das Programm createdb.....	38
Datenbanken löschen.....	40
Backup der Datenbank.....	40
Das Programm pg_dump.....	41
Das Programm pg_dumpall.....	45
Restore der Datenbank.....	47
Mit dem Programm psql.....	47
Das Programm pg_restore.....	48
Plattenplatz optimieren.....	50
Routine Reindexing.....	50
Migration der Datenbank / Einführung einer neuen Version.....	51
Datenbank Aktivitäten überwachen.....	53
Version 8 – Postgres und Windows.....	54
Installation.....	54
Das Programm psql konfigurieren.....	63
Arbeiten mit der Datenbank.....	65
Deinstallation.....	66
Client-Software und andere Zugriffsmöglichkeiten.....	67
Installation der Programme psql und unixODBC auf einem LINUX-Host.....	68
SuSE-Distribution.....	68
Red Hat/Fedora Distribution.....	73

Der ODBC-Treiber für Windows-Systeme.....	75
Das Programm psql.....	76
Allgemeines zum Programm psql	76
Die Terminaloptionen des Programms psql.....	79
Verwalten von Benutzern, Gruppen und Rechte.....	82
Benutzer verwalten.....	82
SQL-Befehle.....	82
Programm createuser.....	83
Programm dropuser.....	85
Gruppe verwalten.....	86
Zugriffsrechte/Privilegien verwalten.....	86
Anhang A - SQL-Befehle.....	88
Anhang B – Anmerkung zum Datenbankentwurf.....	93
Datentypen.....	93
Schemas.....	95
Anhang C - weitere Möglichkeiten zur Administration der Datenbank.....	96
Administration über das Programm telnet.....	96
SuSE-LINUX und telnet.....	97
Red Hat/Fedora LINUX und telnet.....	99
Mit Telnet arbeiten.....	102
Administration über das Programm EMS PostgreSQLManager.....	103
Administration über das Programm pgAdminIII.....	104
Anhang D – die Zeichensätze.....	106
Anhang E - Liste der Dienstprogramme.....	107
Quellen.....	108

Die ersten Schritte in der Administration der PostgreSQL-Datenbank

Diese Seite bleibt aus technischen Gründen frei

Vorwort

Weshalb das Buch entstand:

Durch meine Beschäftigung mit JAVA, C, C++ und LINUX befasste ich mich mit der Datenbank PostgreSQL näher. Da es zu diesem Zeitpunkt relativ wenig deutsche Literatur über diese Datenbank gab, schrieb ich den Leitfaden „Die ersten Schritte in der Administration mit der PostgreSQL-Datenbank“. Mit der Zeit wurde der Leitfaden immer umfangreicher und rückte auch mehr von der Administrationsseite ab.

Hinweis über das Buch:

1. Dieses Buch beschreibt den Einstieg in den Umgang mit der PostgreSQL-Datenbank. Diese Beschreibung wendet sich an die Leser, die bisher wenig mit Linux gearbeitet haben. Das Buch eignet sich auch für die Gelegenheitstäter, die sich in großen Abständen mit dem Thema befassen. Sie erhalten übrigens die Originalunterlagen von der Web-Seite www.postgresql.org (sie sind hier in Englisch abgefasst), oder auch in Ihrer Linux-Distribution (teilweise mit deutscher Beschreibung).
2. Der Installations- und Einrichtungsvorgang ist aus Sicht der SuSE-Linux Professional Distribution 8.0 beschrieben. Auf andere Distributionen wird aber eingegangen. Bei den nachfolgenden und vorhergehenden Versionen kann es Abweichungen geben.
3. Der Leser installiert und bedient die Software auf eigenes Risiko. Der Autor lehnt jede Haftung ab.
4. Über mögliche Anregungen und Hinweise würde ich mich sehr freuen (eMail: stefan@kunick.org).
5. Die in diesem Buch verwendeten Soft- und Hardwarebezeichnungen sind in vielen Fällen auch eingetragene Warenzeichen; sie werden ohne Gewährleistung der freien Verwendbarkeit benutzt. Ich richte mich im Wesentlichen nach der Schreibweise der Hersteller. Die Wiedergabe von Waren- und Handelsnamen berechtigt nicht zur Annahme, dass solche Namen (im Sinne der Warenzeichen und Markenschutz-Gesetzgebung) als frei zu betrachten sind.

Dank an:

Erika Aupperle, Cornelia Boenigk und Janny Kunick für die Durchsicht des Buches

Waiblingen, den 20.02.2005

Stefan Kunick

Über die Datenbank PostgreSQL

PostgreSQL ist eine relationale Datenbank, die auf der POSTGRES Version 4.2 basiert. Sie wurde an der Universität von Kalifornien am Berkeley Computer Science Institut entwickelt. Das Projekt POSTGRES DBMS begann im Jahr 1986. 1994 fügten Andrew Yu und Jolly Chen den SQL Interpreter hinzu. Die Version Postgres95 wurde im Web veröffentlicht und war der Nachfolger des POSTGRES Berkeley Codes. Als Open-Source-Programm steht es frei verfügbar unter der Adresse <http://www.postgresql.org> im Internet. Es unterstützt die SQL-Standards 92 und 99.

Die Größe einer Datenbank ist unbegrenzt und eine Tabelle kann bis zu 64 Terra-Byte groß werden. Bei der Anzahl der Datensätze gibt es keine Grenzen (der verfügbare Plattenplatz einmal ausgenommen). Die Anzahl der Spalten ist auf 1600 begrenzt und jede Spalte nimmt maximal Daten bis zu einem Gigabyte auf.

Zusätzlich kann man auch Bilder in der Datenbank abspeichern. Ein weiterer Pluspunkt der Datenbank sind die Funktionen. Mit ihnen kann man direkt SQL-Befehle auf der Datenbank ausführen.

Installation der Datenbank auf der LINUX-Host

Verschiedene LINUX-Distributoren, wie z.B. SuSE und Red Hat bieten Ihnen auf Ihren CD's, bzw. DVD's auch eine Postgres-Version an. Das Verfahren ist in der Anfangsphase im Umgang mit der Datenbank recht einfach und Sie kommen hier auch ohne einen Download vom Internet aus. Leider sind diese Versionen nicht immer die aktuellsten. Ein Punkt gegen eine Installation von den Datenträgern der LINUX-Distributoren ist der, dass die Programme dann im Verzeichnis /usr/bin abgelegt werden. Bei einem möglichen Update auf eine höhere Version (die Sie sich vielleicht aus dem Internet heruntergeladen haben) müssen Sie dann die Programme aus diesem Verzeichnis entfernen.

(Anmerkung des Autors: ab der Version 8.0 benötigen Sie das Programm CYGWIN, bei einer Installation unter Windows nicht mehr).

Die Installation der Datenbank läuft in mehreren Stufen ab. Dazu können Sie folgende Wege einschlagen:

Stufe	LINUX-Distribution egal	SuSE-Distribution	Red Hat/Fedora-Distribution	Windows
Installation der Serversoftware	Siehe Kapitel: Installation einer neuen Version aus dem Internet	Siehe Kapitel: Installation mit der SuSE-Distribution	Siehe Kapitel: Installation mit der Red Hat-Distribution	Siehe Kapitel: Installation unter Windows
User postgres	Siehe Kapitel: Der User postgres (SuSE)	Siehe Kapitel: Der User postgres (SuSE)	Siehe Kapitel: Der User postgres (Red Hat)	
Datenbank einrichten	Siehe Kapitel: Weiterführende Arbeiten			

Installation einer neueren Version aus dem Internet

Die hier beschriebenen Arbeitsschritte müssen Sie entweder als Benutzer root oder mit Administrator Rechten durchführen (auf dem PC hatte ich LINUX mit den Softwarepaketen für eine Arbeitsstation installiert und dem KDE-Desktop).

Schritt	Inhalt
1	<p>Über Yast (oder bei anderen LINUX-Distributionen ähnliches Programm (bei Fedora z.B. Hinzufügen/Entfernen von Applikationen)) folgende Softwarepakete installieren:</p> <ul style="list-style-type: none"> - gcc - cpp - make <p>ANMERKUNG zu SuSE 8.0: Hier finden Sie die Pakete unter den Rubriken Entwicklung/Programmiersprachen c und c++ und Entwicklung/Tools/Building</p> <p>ANMERKUNG zu SuSE 8.2: Sie finden die Pakete unter c/c++ Compiler und Werkzeuge und Entwicklung erweitert. Aus dem Paket Entwicklung erweitert installieren Sie noch die verschiedenen Librarys (z.B. Readline)</p> <p>ANMERKUNG zu SuSE 9.1: Setzen Sie den Filter bei der Paketauswahl auf Selektionen. Dann finden Sie den Compiler unter dem Eintrag C/C++Compiler und Werkzeuge (Es reichen die Pakete: gcc, readline-devel (Filter auf Paketgruppen setzen und unter Entwicklung, Bibliotheken, c und c++ suchen), zlib-devel (suchen wie bei readline-devel) und make).</p> <p>ANMERKUNG zu RedHat 9/Fedora: Sie installieren hier die Entwicklungstools aus dem Bereich der Softwareentwicklung</p>
2	<p>Wenn Sie die Software von einer CD in ein Verzeichnis kopieren, dann mounten Sie die CD über den Befehl mount /media/cdrom. Die Daten der CD finden Sie im Verzeichnis /cdrom (vorher allerdings in das CD-Rom-Laufwerk einlegen). Bei den neueren Versionen ist das teilweise nicht mehr nötig.</p>
3	<p>Mit dem Befehl mkdir /usr/local/src/Postgres legen Sie ein neues Verzeichnis für die Quellen an</p>
4	<p>Kopieren Sie die gz-Datei (z. B. Postgresql-7.4.3.tar.gz, die Zahlenkombination 7.4.3 steht hier für die Version 7.4.3, bei neueren Versionen ändert sie sich) in das neu angelegte Verzeichnis (z.B. mit dem cp-Befehl)</p>

Schritt	Inhalt
5	Wenn Sie im Arbeitsschritt 2 eine CD gemountet haben, so können Sie jetzt die CD mit dem Befehl <i>umount /media/cdrom</i> unmounten. Anschließend entnehmen Sie die CD.
6	Jetzt entkomprimieren Sie die Datei mit dem Befehl <i>gunzip postgresql-7.4.3.tar.gz</i>
7	Danach packen Sie die notwendigen Dateien aus. Geben Sie den Befehl <i>tar xf postgresql-7.4.3.tar</i> ein.
8	Wechseln Sie nun in das Verzeichnis postgresql-7.4.3 mit dem Befehl <i>cd postgresql-7.4.3</i>
9	Als nächstes konfigurieren Sie den Source-Tree. Er enthält die Optionen, mit denen Sie PostgreSQL übersetzen. Sie geben dazu den Befehl <i>./configure</i> ein (sollten Sie die Library Readline nicht installiert haben, so können Sie diese Library auch weglassen. Geben Sie dazu den Befehl <i>./configure --without-readline</i> ein. Diese Library ist für die Befehls-History zuständig). Wenn alles funktioniert hat, erhalten Sie die Datei config.status. Wenn Sie möchten, dann können Sie auch die Meldungen in Deutsch ausgeben lassen. Sie benötigen nur den Parameter <code>--enable-nls='de'</code> .
10	Mit dem Build-Prozess erzeugen Sie nun die ausführbaren Dateien. Der Prozess kann längere Zeit den Rechner in Anspruch nehmen. Das Kommando lautet <i>gmake</i> . Ob das Programm erfolgreich durchlief sehen Sie durch die Meldung: <i>All of PostgreSQL is successfully made. Ready to install.</i>
11	Mit einem Regression-Test testen Sie den soeben erzeugten Build. Diese Aktion können Sie nicht als Benutzer root durchführen. Vergeben Sie dazu die notwendigen Rechte an einen anderen Benutzer. Die Befehle sehen so aus: 1. <i>chmod -R a+w src/test/regress</i> 2. <i>chmod -R a+w contrib/spi</i> 3. <i>su <anderer Benutzer></i> 4. <i>gmake check</i> (Sie erhalten dann die Meldung <i>All 93 tests passed</i>) 5. wieder als Administrator anmelden

Schritt	Inhalt
12	Mit dem Befehl <i>gmake install</i> installieren Sie PostgreSQL in das Verzeichnis, das in der Datei <i>configure</i> angegeben wurde (Standard: /usr/local/pgsql/bin, /usr/local/pgsql/lib, siehe Eintrag <i>ac_default_prefix</i> , Vorsicht: es gibt mehrere Einträge). Wenn Sie die Datei mit dem Editor <i>vi</i> bearbeiten, dann können Sie nach den Einträgen mit <i>/ac_default_prefix</i> suchen. (TIPP: Rechte für die Anwender überprüfen und vergeben)
13	Setzen Sie nun die Umgebungsvariablen für die Shared Libraries (TIPP: am besten im Startskript vermerken. Z.B. in der Datei /etc/profile (dann gelten die Werte für alle)). Das Vorgehen kann bei anderen Linux Distributionen ggf. abweichen. Die Befehle lauten: 1. <i>LD_LIBRARY_PATH=/usr/local/pgsql/lib</i> 2. <i>export LD_LIBRARY_PATH</i>
14	Passen Sie danach die Umgebungsvariable PATH an . 1. <i>PATH=/usr/local/pgsql/bin:\$PATH</i> 2. <i>export PATH</i>
15	Die Umgebungsvariable MANPATH ist für die Manpages wichtig. Die Anweisungen lauten: 1. <i>MANPATH=/usr/local/pgsql/man:\$MANPATH</i> 2. <i>export MANPATH</i> (der Tipp vom Schritt 14 gilt hier übrigens auch) ANMERKUNG: Wenn Sie auf die Umgebungsvariablen später nochmals zugreifen möchten, müssen Sie sie dann neu setzen (siehe Kapitel Weiterführende Arbeiten). Das ist dann der Fall, wenn Sie sich als normaler Anwender angemeldet haben und die Installation als User „su“ durchführen und auch die Umgebungsvariablen unter User „su“ gesetzt haben.
16	Nun geht es je nach Distribution entweder im Kapitel <i>Der User postgres (SuSE)</i> , <i>Der User postgres (RedHat/Fedora)</i> weiter.

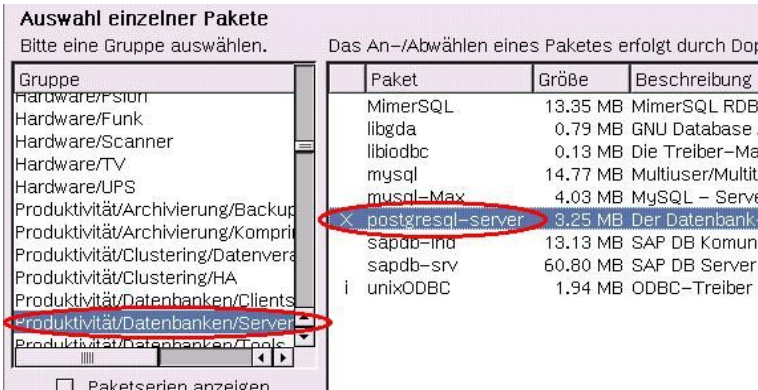
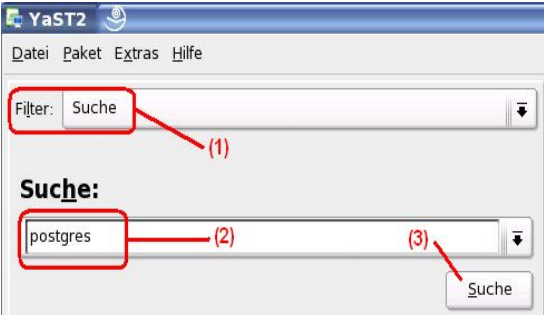
Installation auf einer Host mit SuSE-LINUX

Installation mit der SuSE-Distribution

Wie schon am Anfang beschrieben, benötigen Sie die Professional Version von Linux. Sie enthält auch die entsprechende Software. Sie können natürlich auch alternativ eine Version aus dem Internet herunterladen und


auf dem Rechner installieren.



Schritt	Inhalt / Aktion
1	Rufen Sie als erstes das YaST2-Kontrollzentrum auf. Wenn Sie den KDE-Desktop einsetzen drücken Sie hier die Start -Taste, Einstellungen , Yast menu und dann Yast Kontrollzentrum .
2	<p>Unter der Rubrik Software (linke Seite) finden Sie das ICON Software installieren/löschen. Bitte das Programm Software installieren/löschen mit einem Doppelklick aufrufen.</p> <p><i>Abbildung 1 SuSE YaST</i></p>

Schritt	Inhalt / Aktion
	<p>Als nächstes installieren Sie die „Server-Anwendung“ auf ihrem Linux-Host. Aus dem Paket Produktivität/Datenbanken/Server wählen Sie das Paket postgresql-server aus (drücken Sie hier einfach den Knopf Aus/Abwählen).</p>
3	<div></div> <p>Abbildung 2 SuSE Pakete auswählen</p>
	<p>ANMERKUNG zu neueren Versionen: Stellen Sie den Filter auf Suchen (siehe (1))ein und geben Sie in dem Eingabefeld den Begriff postgres (siehe (2)) ein. Wenn Sie dann den Knopf Suche drücken, erhalten Sie eine komplette Trefferliste aller Postgres-Pakete.</p>
4	<div></div> <p>Abbildung 3 SuSE optimale Suche nach Paketen</p>
	<p>Soll auf der Linux - Host nur die Datenbank laufen dann können Sie schon die Auswahl abschließen und die geforderte CD einlegen.</p>

Schritt	Inhalt / Aktion
5	Wenn die Software installiert ist, gehen Sie weiter zu dem Kapitel <i>Der User postgres (SuSE)</i> .

Der User postgres (SuSE)

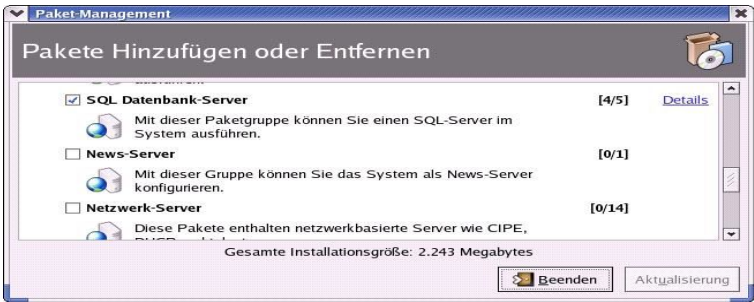
Schritt	Inhalt / Aktion
1	<p>Überprüfen Sie, ob der Anwender postgres schon in der Benutzerverwaltung vorhanden ist (drücken Sie die Start-Taste, System, Konfiguration, KUser (Benutzerverwaltung) (In den neueren SuSE-Versionen müssen Sie das Programm KUser noch installieren. Sie finden es im Paket kadmin3kde). (TIPP: das Programm Yast geht auch)</p> <div data-bbox="272 754 1023 1137"></div> <p>Abbildung 4 SuSE Benutzerverwaltung</p> <p>ANMERKUNG zu SuSE 9.1: Sollten Sie eine Version aus dem Internet installieren, dann ist der User postgres noch nicht vorhanden.</p>

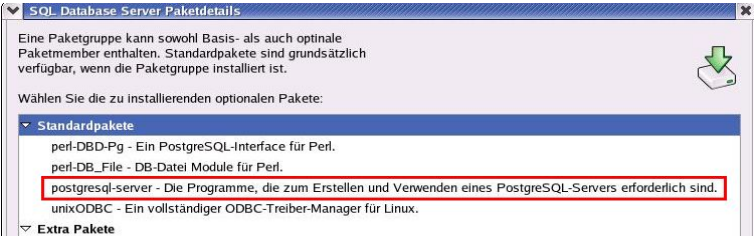
Schritt	Inhalt / Aktion
2	<p>Als Nächstes überprüfen Sie noch einige Einstellungen des Benutzers postgres und vergeben ihm auch ein Passwort! Drücken Sie dazu den Knopf Edit in der Menueleiste. Anschließend erhalten Sie den Zugriff auf die Benutzereigenschaften. Über den Knopf Passwort setzen ändern Sie das Passwort ab.</p>  <p><i>Abbildung 5 SuSE Benutzer anlegen</i></p>
3	<p>Zum Schluss überprüfen Sie noch, ob der Anwender zur Gruppe daemon gehört. Über den Karteireiter Gruppen erreichen Sie die Seite.</p>  <p><i>Abbildung 6 SuSE Benutzer und Gruppe</i></p>
4	<p>Alle folgenden Schritte finden Sie im Kapitel Weiterführende Arbeiten.</p>

Installation auf einer Host mit Red Hat/Fedora-LINUX

Installation mit der Red Hat/Fedora-Distribution

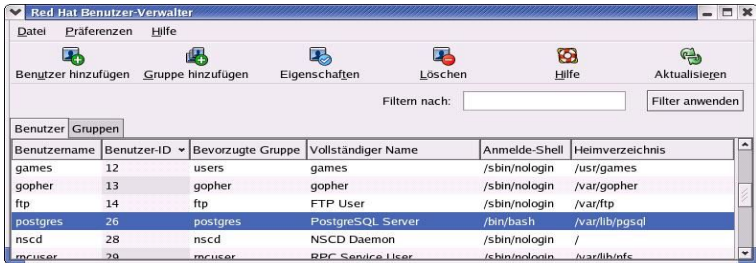
Für die Datenbank verwendete ich die Version 9 von Red Hat (die Version mit den 3 CD's. Sie können die Version unter www.linuxiso.org auch herunterladen). Red Hat hat diese Produktlinie unter dem Namen Fedora ausgegliedert. Von der o.g. Homepage können Sie sich die Software auch downloaden. Alternativ bietet Ihnen der Zeitschriftenhandel die Distribution auch an (Kostenpunkt ca. 10 Euro). Von der Gestaltung her hat sich nicht viel geändert.


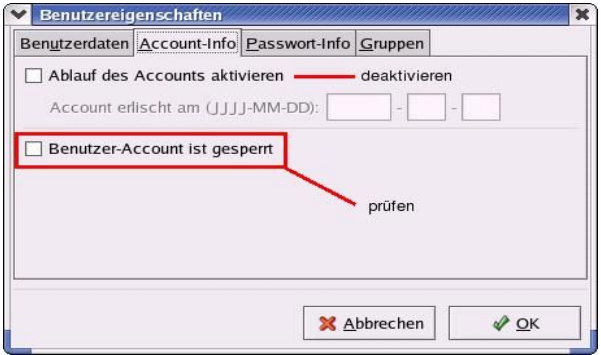
Schritt	Inhalt / Aktion
1	Drücken Sie die Start-Taste (mit dem roten Hut). Danach geht es weiter mit den Menüpunkten Systemeinstellungen und Hinzufügen / Entfernen von Applikationen
2	<p>Im Fenster Pakete Hinzufügen oder Entfernen finden Sie den Eintrag SQL Datenbank-Server. Wählen Sie diesen Punkt aus.</p> <div data-bbox="266 799 1020 1102"></div> <p>Abbildung 7 Red Hat Pakete Hinzufügen oder Entfernen Red Hat</p> <p>Bei Fedora finden Sie die Pakete unter der Rubrik Server und dem Eintrag PostgreSQL Datenbank.</p>

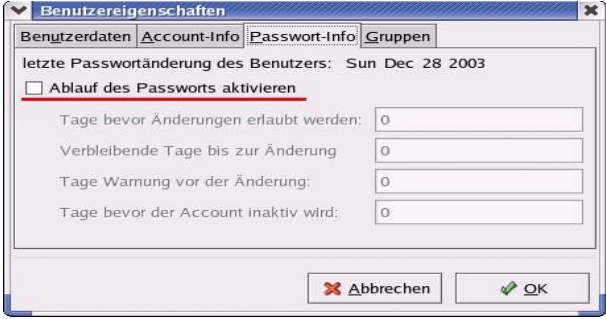
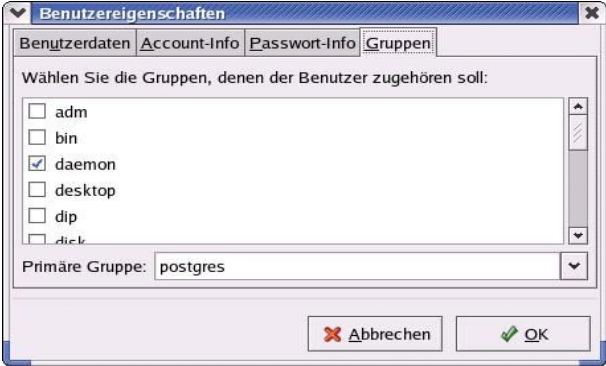
Schritt	Inhalt / Aktion
3	<p>Wenn Sie im vorherigen Schritt auf den Eintrag Details drücken, sehen Sie im Detail, welche Software mit den Standardpaketen installiert wird.</p>  <p>Abbildung 8 Red Hat Paketdetails Red Hat</p>
4	<p>Danach schließen Sie die Auswahl ab, drücken den Knopf Aktualisierung und legen die geforderten CD's ein.</p>
5	<p>Wenn die Software installiert ist, gehen Sie weiter zu dem Kapitel <i>Der User postgres (Red Hat/Fedora)</i>.</p>

Der User postgres (Red Hat/Fedora)

Wie auch bei SuSE-Distribution , überprüfen wir nun die Einstellungen des Benutzers *postgres*.

Schritt	Inhalt / Aktion																																										
1	Drücken Sie die Start-Taste (mit dem roten Hut). Danach geht es weiter mit den Menüpunkten Systemeinstellungen und Benutzer und Gruppen .																																										
2	<p>Verändern Sie zuerst die Einstellungen über die Darstellung der Benutzer und Gruppen. Sie finden den Punkt im Menüpunkt Präferenzen (Systembenutzer und -gruppen filtern). Wählen Sie anschließend den Benutzer postgres aus und drücken den Knopf Eigenschaften.</p> <div><table><tr><th>Benutzername</th><th>Benutzer-ID</th><th>Bevorzugte Gruppe</th><th>Vollständiger Name</th><th>Anmelde-Shell</th><th>Heimverzeichnis</th></tr><tr><td>games</td><td>12</td><td>users</td><td>games</td><td>/sbin/nologin</td><td>/usr/games</td></tr><tr><td>gopher</td><td>13</td><td>gopher</td><td>gopher</td><td>/sbin/nologin</td><td>/var/gopher</td></tr><tr><td>ftp</td><td>14</td><td>ftp</td><td>FTP User</td><td>/sbin/nologin</td><td>/var/ftp</td></tr><tr><td>postgres</td><td>26</td><td>postgres</td><td>PostgreSQL Server</td><td>/bin/bash</td><td>/var/lib/pgsql</td></tr><tr><td>nscd</td><td>28</td><td>nscd</td><td>NSCD Daemon</td><td>/sbin/nologin</td><td>/</td></tr><tr><td>mcuser</td><td>29</td><td>mcuser</td><td>RPC Service User</td><td>/sbin/nologin</td><td>/var/lib/nfs</td></tr></table></div> <p>Abbildung 9 Red Hat Benutzerverwaltung</p>	Benutzername	Benutzer-ID	Bevorzugte Gruppe	Vollständiger Name	Anmelde-Shell	Heimverzeichnis	games	12	users	games	/sbin/nologin	/usr/games	gopher	13	gopher	gopher	/sbin/nologin	/var/gopher	ftp	14	ftp	FTP User	/sbin/nologin	/var/ftp	postgres	26	postgres	PostgreSQL Server	/bin/bash	/var/lib/pgsql	nscd	28	nscd	NSCD Daemon	/sbin/nologin	/	mcuser	29	mcuser	RPC Service User	/sbin/nologin	/var/lib/nfs
Benutzername	Benutzer-ID	Bevorzugte Gruppe	Vollständiger Name	Anmelde-Shell	Heimverzeichnis																																						
games	12	users	games	/sbin/nologin	/usr/games																																						
gopher	13	gopher	gopher	/sbin/nologin	/var/gopher																																						
ftp	14	ftp	FTP User	/sbin/nologin	/var/ftp																																						
postgres	26	postgres	PostgreSQL Server	/bin/bash	/var/lib/pgsql																																						
nscd	28	nscd	NSCD Daemon	/sbin/nologin	/																																						
mcuser	29	mcuser	RPC Service User	/sbin/nologin	/var/lib/nfs																																						

Schritt	Inhalt / Aktion
3	<p>Als Nächstes können Sie unter der Registrierkarte Benutzerdaten dem Anwender ein Passwort vergeben.</p>  <p><i>Abbildung 10 Red Hat Benutzereigenschaften Benutzerdaten</i></p>
4	<p>Unter der Registrierkarte Account-Info entsperren Sie den Benutzer-Account und prüfen, ob eine Verfallszeit vorgesehen ist.</p>  <p><i>Abbildung 11 Red Hat Benutzereigenschaften Account-Info</i></p>

Schritt	Inhalt / Aktion
5	<p>Wechsel Sie danach auf die Karteikarte Passwort-Info. Beachten Sie hier den Ablauf des Passwortes.</p>  <p><i>Abbildung 12 Red Hat Benutzereigenschaft Passwort</i></p>
6	<p>Zuletzt überprüfen Sie noch die Gruppenzugehörigkeit. Mit einem Kreuz bei dem Eintrag daemon können Sie dann die Datenbank als „Dienst“ laufen lassen. Danach geht es weiter im Kapitel Weiterführende Arbeiten</p>  <p><i>Abbildung 13 Red Hat Benutzereigenschaft Gruppen</i></p>

Weiterführende Arbeiten

Die hier beschriebenen Arbeitsschritte sind alle unabhängig von der entsprechenden LINUX-Distribution. Wir richten hier die Datenbank ein.

Schritt	Inhalt / Aktion
1	<p>Melden Sie sich als Anwender root an (von der Konsole mit dem Befehl su geht es auch) und wechseln Sie in das Verzeichnis /var/lib/pgsql. Legen Sie dort das Verzeichnis data an (INFO: eine Übersicht des Dateibaumes finden Sie im Anhang).</p> <p>ANMERKUNG: Sie können natürlich auch ein anderes Verzeichnis verwenden. Später können Sie auch das data-Verzeichnis in ein anderes Verzeichnis verschieben. Beim Programmstart geben Sie dann das geänderte Verzeichnis mit an. (Sie können auch ein über NFS gemapptes Verzeichnis verwenden, es belastet dann aber das Netz und die Kapazität des Netzes ist auch entscheidend). Übrigens, wenn Sie sich als Anwender postgres anmelden, dann können Sie ohne Probleme in Ihrem Home-Directory das Verzeichnis data anlegen und sparen sich den nächsten Arbeitsschritt.</p>
2	<p>Mit dem Befehl chown wechseln Sie den Besitzer des Verzeichnisses /var/lib/pgsql und /var/lib/pgsql/data. Der genaue Syntax lautet: chown -c postgres pgsql (aus dem Verzeichnis /var/lib heraus) bzw. chown -c postgres data (aus dem Verzeichnis /var/lib/pgsql).</p>

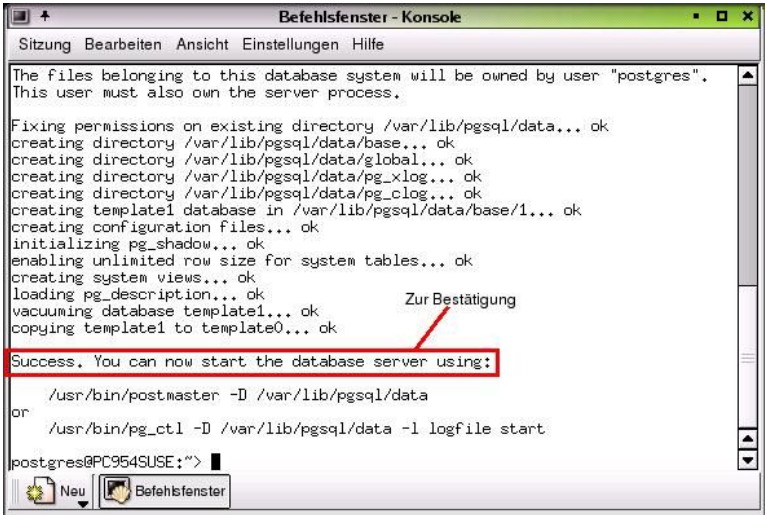
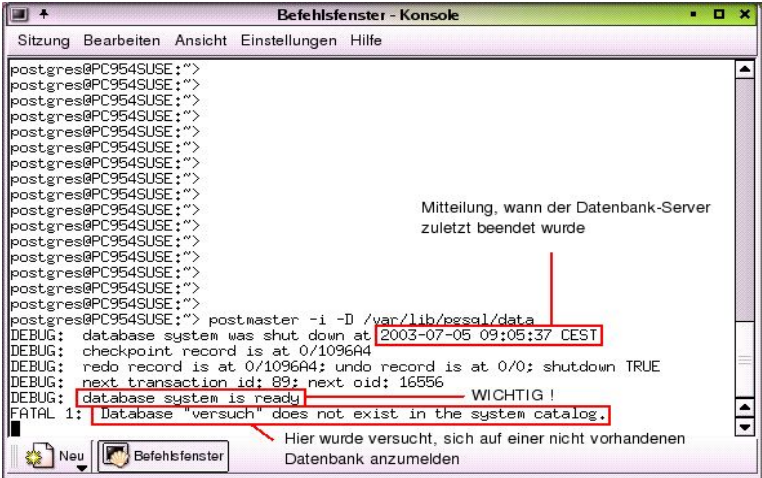
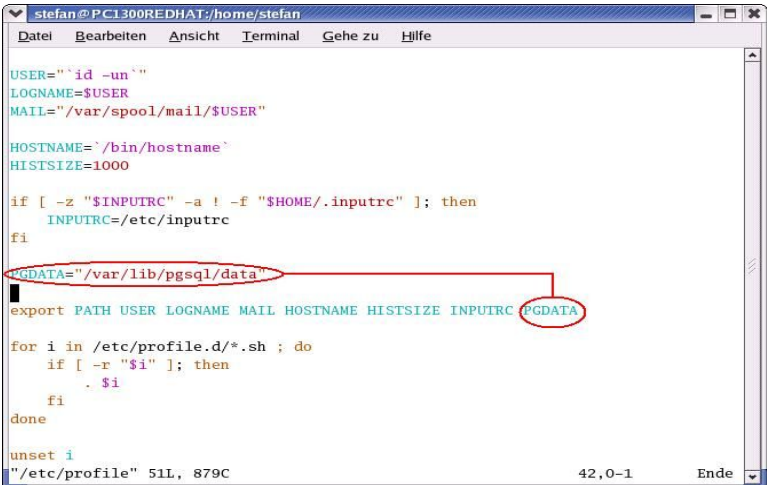
Schritt	Inhalt / Aktion
3	<p>Melden Sie sich anschließend als User postgres an (geht auch über die Konsole mit dem Befehl su postgres vom aktuell angemeldeten Benutzer aus) und starten Sie die Datenbank mit dem Befehl: initdb -D /var/lib/pgsql/data</p>  <p>The files belonging to this database system will be owned by user "postgres". This user must also own the server process.</p> <p>Fixing permissions on existing directory /var/lib/pgsql/data... ok creating directory /var/lib/pgsql/data/base... ok creating directory /var/lib/pgsql/data/global... ok creating directory /var/lib/pgsql/data/pg_xlog... ok creating directory /var/lib/pgsql/data/pg_clog... ok creating template1 database in /var/lib/pgsql/data/base/1... ok creating configuration files... ok initializing pg_shadow... ok enabling unlimited row size for system tables... ok creating system views... ok loading pg_description... ok vacuuming database template1... ok copying template1 to template0... ok</p> <p>Success. You can now start the database server using:</p> <p>or</p> <p>/usr/bin/postmaster -D /var/lib/pgsql/data</p> <p>/usr/bin/pg_ctl -D /var/lib/pgsql/data -l logfile start</p> <p>postgres@PC954SUSE:~\$</p>

Abbildung 14 Datenbank einrichten

Nähere Erläuterungen zum Start der Datenbank erhalten Sie auf den folgenden Seiten.

Schritt	Inhalt / Aktion
4	<p>Nun können Sie die Datenbank starten mit der Anweisung: postmaster -i -D /var/lib/pgsql/data (mit diesem Prozess läuft dann die Datenbank, beim Start des Systems muss dieser Befehl dann immer eingegeben werden) WICHTIG: zusätzlich mit der Option -i starten, damit es mit der TCP/IP-Verbindung klappt (die Option -i muß vor dem -D kommen). Wenn alles in Ordnung ist erhalten Sie die Meldung database system is ready (siehe unten). Weitere Informationen über das Programm postmaster finden Sie im Kapitel postmaster.</p>  <p style="text-align: center;">Abbildung 15 Datenbank starten</p>
5	<p>Als Nächstes legen Sie eine Datenbank an. Den dazu notwendigen Befehl geben Sie über die Konsole ein (hier sollten Sie als Benutzer postgres angemeldet sein). Er lautet: createdb -E LATIN1 [NAME der Datenbank] (z.B. createdb -E LATIN1 test) VERWEIS: Siehe auch im Anhang A SQL-Befehle. WICHTIG: Mit der Option -E LATIN1 haben Sie später keine Probleme mit den Umlauten.</p>
6	<p>Testweise können Sie jetzt die Datenbank aufrufen (Sie müssen noch als Anwender postgres angemeldet sein, sonst erhalten Sie kein Zugriffsrecht auf die Datenbank). Die Verbindung zur eben angelegten Datenbank nehmen Sie über den Befehl psql -d [NAME der Datenbank] auf. Das Programm beenden Sie mit der Anweisung q. Eine nähere Beschreibung des Programms psql erhalten Sie im Kapitel psql.</p>

Schritt	Inhalt / Aktion
7	<p>Im unserem Fall lautet das Data - Verzeichnis des SQL-Servers /var/lib/pgsql/data. In ihm finden Sie die wichtigsten Konfigurationsdateien. Um z.B. von anderen Computern auf die Datenbank unserer LINUX-Host zugreifen zu können, müssen Sie die Datei pg_hba.conf bearbeiten (siehe extra Kapitel)! Eine weitere wichtige Konfigurationsdatei ist die Datei postgresql.conf. Sie enthält z.B. die Anzahl der maximal möglichen Verbindungen und ob z.B. eine TCP/IP Verbindung möglich ist (siehe auch später). In der Datei postmaster.pid finden Sie unter anderem die Process-ID des zurzeit gestarteten Programmes postmaster (WICHTIG: um die Datenbank zu beenden).</p>
8	<p>TIPP: Damit Sie nicht immer das Datenverzeichnis der Datenbank eingeben müssen empfiehlt sich der Einsatz der Systemvariablen PGDATA. Ich habe dazu die Variable PGDATA in die Datei etc/profile eingetragen und anschließend exportiert. Nach einem erneuten abarbeiten der Datei (z.B. bei einem Systemstart) kennt dann das System die Variable. Bitte prüfen Sie vorher, ob die Variable nicht schon in einem anderen Skript gesetzt wurde. Übrigens, wenn Sie eine Version aus dem Internet installiert haben, dann können Sie auch die anderen Variablen hier berücksichtigen (PATH, MANPATH) (ANMERKUNG: Bei SuSE 9.1 wird empfohlen die Einträge in der Datei /etc/profile.local vorzunehmen).</p>  <pre> stefan@PCI300REDHAT:/home/stefan Datei Bearbeiten Ansicht Terminal Gehe zu Hilfe USER="`id -un`" LOGNAME=\$USER MAIL="/var/spool/mail/\$USER" HOSTNAME=`bin/hostname` HISTSIZE=1000 if [-z "\$INPUTRC" -a ! -f "\$HOME/.inputrc"]; then INPUTRC=/etc/inputrc fi PGDATA="/var/lib/pgsql/data" export PATH USER LOGNAME MAIL HOSTNAME HISTSIZE INPUTRC PGDATA for i in /etc/profile.d/*.sh; do if [-r "\$i"]; then . \$i fi done unset i "/etc/profile" 51L, 879C 42,0-1 Ende </pre> <p>Abbildung 16 PGDATA</p>

Die Datenbank kann man übrigens auch automatisch beim Hochfahren des Systemes starten. Beachten Sie dabei, dass dann der Benutzer **postgres** kein Passwort enthalten darf (**VORSICHT!**)! Um die Datenbank zu starten, bearbeiten Sie im Verzeichnis /etc die Datei rc.local (**ANMERKUNG:** Diese Datei hat die gleichen Aufgaben wie die Datei AUTOEXEC.BAT bei den Betriebssystem Windows 98/95). Das Prinzip wurde verbessert. Beim Systemstart durchläuft LINUX verschiedene Runlevels. In jedem Runlevel startet das System verschiedene Skripts in unterschiedlichen Verzeichnissen (siehe auch die README-Datei im Verzeichnis /etc/rc.d). Sie finden die Verzeichnisse für die einzelnen Runlevels unter dem Verzeichnis /etc/rc.d). Die Nummer des Runlevels ist mit im Verzeichnisnamen kodiert (rc[Runlevel].d). Erst ab dem Runlevel 3, wenn das Netzwerk gestartet ist, macht es Sinn die Datenbank zu starten. Beachten Sie, dass nur Skriptdateien berücksichtigt werden, die mit S oder K beginnen und eine zweistellige Zahl enthalten (das S steht für Start, das K für Kill). Kopieren Sie sich die Datei /etc/init.d/skeleton in das entsprechende Verzeichnis und bearbeiten die Datei (**besser:** Sie kopieren die Datei in das Verzeichnis **/etc/init.d** und legen einen symbolischen Link in das Verzeichnis. Der Link in dem Verzeichnis darf bzw. muss dann einen anderen Namen haben. Mit dem Befehl **ln -s <quelle> <ziel>** legen Sie den Link an. Mit dem Befehl **ls -l** sehen Sie Ihre „Quelldatei“).

ANMERKUNG: das Verzeichnis rc.d ist nur ein Link auf ein anderes Verzeichnis. Beim Systemstart ruft LINUX alle Programme und Skripts als Superuser auf. Als Superuser, oder als anderer Benutzer, ist es allerdings nicht zulässig das Programm postmaster zu starten. Deshalb rufen Sie einen Daemon-Prozess auf und geben als Anwendernamen des Benutzers postgres an. Das vereinfachte Skript sieht dann wie unten gezeigt aus (Sie können in das Verzeichnis /etc/rc.d/rc5.d eine Verknüpfung legen) (Übrigens: Das Byte zur Ausführung der Datei nicht vergessen! In der ersten Zeile der Datei sollte auch der Eintrag **#!/bin/sh** vorhanden sein):

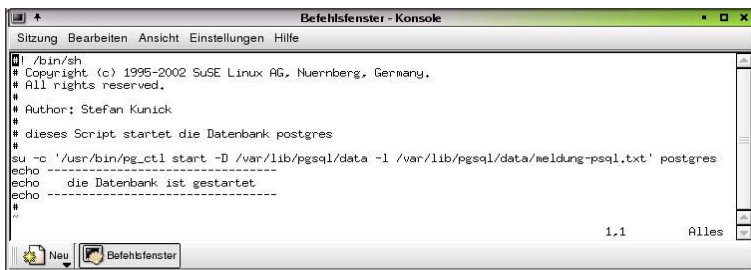


Abbildung 17 Skript zum Start der Datenbank

Die wichtigste Zeile in dem Skript ist die Anweisung ***su -c '/usr/bin/pg_ctl start -D /var/lib/pgsql/data' postgres***. (**ANMERKUNG:** Sollte es Probleme mit dem Login von postgres geben, schreiben Sie anstelle postgres -l postgres. Wenn Sie eine Version aus dem Internet auf der Host installiert haben, dann finden Sie die Datei im Verzeichnis ***/usr/local/pgsql/bin***). Zusätzlich können Sie die Meldungen beim Start auch noch in eine Datei umleiten. Der Daemon pg_ctl startet die Datenbank. Wenn Sie vor dem Datenverzeichnis noch die Option ***-l logfile*** hinzufügen, dann werden alle Meldungen der Datenbank in der Logfile mit protokolliert. Damit auch weitere Anwender über TCP/IP auf die Host zugreifen können, ändern Sie in der Datei /var/lib/pgsql/data/postgresql.conf den Eintrag ***#tcpip_socket = false*** auf:

tcpip_socket = true

ANMERKUNG: Sie können die Datenbank entweder über das Programm ***postmaster*** starten, oder mit ***pg_ctl*** (siehe auch unter dem Kapitel Weitere Dienstprogramme). Sie können auch die Befehle in die Datei ***/etc/rc.d/rc.local*** einbauen. Auch die Datei ***contrib/start-scripts/linux*** (befindet sich im Verzeichnis der Version aus dem Internet, nach dem Entpacken der Dateien). Besonders wichtig ist die Einstellung bezüglich des TCP/IP-Zugriffs. Übergeben Sie die Information entweder per Parameter dem Programm postmaster, oder vermerken Sie es in der Datei postgresql.conf. Das System hat für Sie schon eine erste Datenbank (Name ***template1***) angelegt. Sie dient Ihnen später als Kopiervorlage für weitere Datenbanken.

Wenn Sie bei der Red Hat/Fedora Distribution die Software von der CD installieren, können Sie die Datenbank starten. Rufen Sie das Programm **Dienste** auf (**Start-Taste, Systemeinstellungen, Servereinstellungen**). Wie auch bei der SuSE-Distribution finden Sie die entsprechende Start-Datei im Verzeichnis `/etc/rc.d/rc5.d` (`S85postgresql`). Über das Programm **Dienste** können Sie den Programmstart veranlassen.

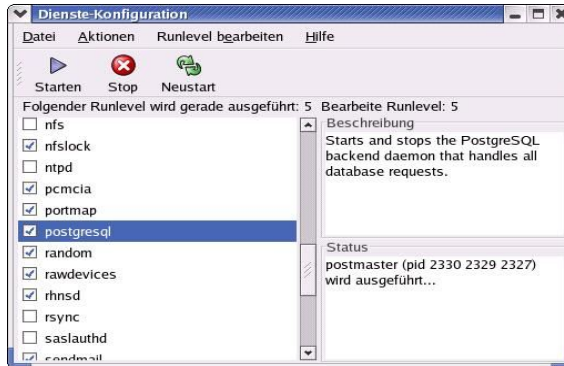


Abbildung 18 Dienst-Konfiguration

Automatisches Login

Legen Sie im Home-Verzeichnis des Users die Datei ***.pgpass*** an. In ihr tragen Sie dann den User-Namen und die Passwörter ein, die dann als default-Wert für den Zugriff auf die Datenbank verwendet werden. Das Format der Einträge sieht dann so aus:

hostname:port:datenbank:benutzername:passwort

Anstelle eines Feldes könne Sie auch das Wildcartzeichen * einsetzen. Den Einträgen mit : und \ stellen Sie das Zeichen \ voran. Achten Sie auch auf die passenden Zugriffsrechte (mit dem Befehl ***chmod 0600 .pgpass*** ändern). Sind die Zugriffsrechte weniger strikt, dann wird die Datei ignoriert.

Die Datenbank beenden

Wenn Sie den Rechner herunterfahren oder neu starten, dann sollten Sie vorher die Datenbank beenden. Wie auch beim Start der Datenbank, gibt es eine Shell zum beenden der Datenbank (Den Prozess mit dem kill-Befehl zu beenden, kann zu Problemen führen, die Prozess ID finden Sie in der Datei /var/lib/pgsql/data/postmaster.pid, oder über den Befehl ps -A). Am besten Sie verwenden hier die die Shell pg_ctl im Verzeichnis /usr/bin (sehen Sie sich die Datei interessehalber einmal im Editor an, die Datei kann auch in einem anderen Verzeichnis liegen).

Der Syntax ist wie folgt aufgebaut:

```
pg_ctl stop [-W] [-D DATADIR] [-s] [-m SHUTDOWN-MODE]
```

Parameter	Inhalt
-W	Optional - Wartet nicht bis alle Operationen ausgeführt sind
-D	Optional - Das Datenverzeichnis ANMERKUNG: Im Skript ist es als Optional angegeben, es ist aber besser man gibt es an.
-s	Optional - gibt nur Fehler aus, keine zusätzlichen Informationen
-m	Optional - Der Shutdown-Modus (siehe auch weitere Tabelle)

SHUTDOWN-MODE	Inhalt
smart	Ende, nachdem sich alle Clientes abgemeldet haben. Neue Verbindungen lässt der Postmaster nicht mehr zu.
fast	Alle Transaktionen werden noch ausgeführt, danach trennt das Programm alle Verbindungen.
immediate	Beendet den Postmaster ohne Rücksicht auf Verluste.

Auf dem nächsten Screenshot sehen Sie die Meldungen beim Beenden der Datenbank.

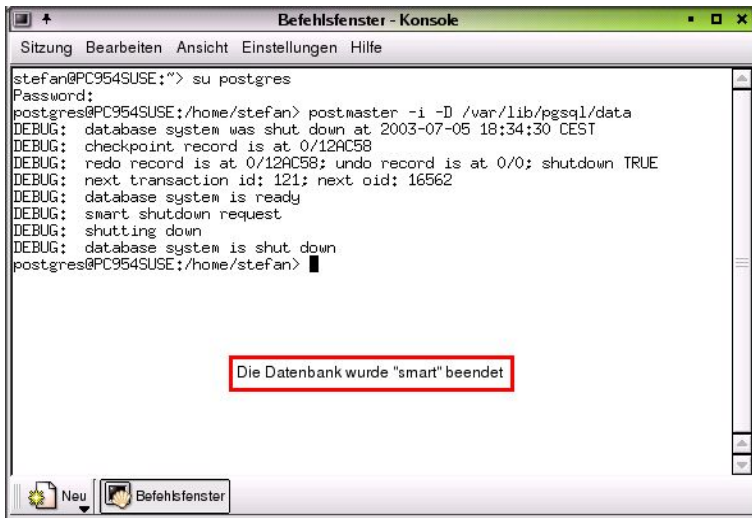


Abbildung 19 Datenbank beenden

WICHTIG: Damit die Datenbank auf jeden Fall beendet wird, sollten Sie ein Shell-Skript in das Verzeichnis `/etc/rc.d/rc0.d` und `/etc/rc.d/rc6.d` legen (am besten ein Link). Es gelten dieselben Bedingungen wie beim Start. Der Befehl sieht dann z.B. so aus:

`su -c '/usr/bin/pg_ctl stop -D /var/lib/pgsql/data -m smart' postgres`

oder bei der Installation aus dem Internet

`su -c '/usr/local/pgsql/bin/pg_ctl stop -D /var/lib/pgsql/data -m smart' postgres`

Die Konfigurationsdatei pg_hba.conf

Ohne diese Datei kann kein anderer LINUX-Host, bzw. Windows-PC auf Ihre Datenbank zugreifen. Sie enthält alle wichtigen Informationen. Diese Datei befindet sich im Verzeichnis PGDATA (z.B. /var/lib/postgresql/data). Folgende Punkte regelt die Konfigurationsdatei:

- 1.) Welcher Host (bzw. welches Netzwerk) darf sich anmelden
- 2.) Welche Zugriffsberechtigung hat jeder Host
- 3.) Welcher Host darf auf welche Datenbank zugreifen

In dieser Steuerdatei werden die Kommentarzeilen mit einem #-Zeichen an der ersten Stelle gekennzeichnet. Leere Zeilen ignoriert das Programm. In einer Konfigurationseinstellung trennen Sie Parameter durch mehrere Leerzeilen oder dem Tab-Zeichen. Jede Einstellung enthält eine IP-Adresse, die Datenbank und den Zugriffsberechtigungstyp. 3 unterschiedliche Möglichkeiten haben Sie (eine Erklärung der Informationen [DBNAME finden Sie in einer weiteren Tabelle]):

Einstellung	Inhalt
host	<p>Hier geben Sie die IP-Adressen der Host oder des Netzwerkes ein (im Beispiel unten das Feld host), die sich auf der Datenbank anmelden dürfen. Das Format sieht so aus:</p> <pre>[host] [DBNAME] [IP_ADDRESS] [ADDRESS_MASK] [AUTH_TYPE]</pre> <p>Die Information ADDRESS_MASK kann auch Platzhalter enthalten. Möchten Sie zum Beispiel allen IP-Adressen aus dem Bereich 192.168.x.x den Zugriff auf die Datenbank erlauben, so geben Sie die ADDRESS_MASK 255.255.0.0 vor. Die Zahl Null steht hier als Platzhalter, an dieser Stelle wird die IP-Adresse nicht geprüft! In neueren Versionen können Sie anstelle der IP-Adresse und der Adressmaske auch nur die CIDR-ADRESSE eingeben. Die Zahl nach dem Schrägstrich steht für die Anzahl von Bits (von rechts nach links) in der IP-Adresse, die für Host reserviert sind (z.B. 127.0.0.1/32 für eine komplette Adresse und 60.0.0.0/24 für einen Host in einem A-Netzwerk).</p> <p>WICHTIG: Sie dürfen hier nur IP-Adressen eingeben und keine Host-Namen!</p>

Einstellung	Inhalt
hostssl	Der Eintrag ist mit der Einstellung host identisch. Die Verbindung wird aber <u>nur</u> über eine sichere SSL IP- Connection aufgenommen. Mit der Einstellung host sind neben den „unsicheren“ auch die sicheren Verbindungen möglich.
local	Über diese Einstellung sind Verbindungen nur über den local UNIX domain socket möglich.

Erklärung:

Feld	Inhalt
DBNAME	Es enthält den Namen einer Datenbank. Die Information all steht für alle Datenbanken. Mit sameuser gestatten Sie den Zugriff auf die Datenbank, die denselben Namen wie der Anwender hat, der sich zurzeit auf der Datenbank anmeldet.
AUTH_TYPE	<p>Die Möglichkeiten sehen so aus:</p> <ul style="list-style-type: none"> - trust => jeder gültige Benutzername wird akzeptiert (Anmerkung: Gefährlich, Anwender anderer Hosts können sich ohne Passwort als User postgres auf der Datenbank anmelden!) - password => hier gibt der Anwender ein Passwort ein. Der Anwender muss nicht unbedingt als gültiger Benutzer der Host angelegt sein. Wenn das Argument nicht verwendet wird, dann sieht das Programm in der Tabelle pg_shadow nach. - md5 => ist wie password aufgebaut, der Anwendername ist verschlüsselt (notwendig für Clients deren Versions-Nr. unter der Version 7.2 liegt). - ident => hier meldet sich der Anwender unter seinem Betriebssystemnamen auf der Datenbank an. Auch bei diesem Verfahren ist Vorsicht geboten. Auf dem Server müssen Sie den Dämon identd laufen lassen. - krb4 => Kerberos V4 Zugriffsberechtigung, bei TCP/IP-Verbindungen, funktioniert nicht bei local UNIX-domain sockets. - krb5 => Kerberos V5 Zugriffsberechtigung (siehe auch krb4) - reject => Diese Einstellung bedeutet, dass diese IP-Adresse zurückgewiesen wird
AUTH_ARGUMENT	Ist teilweise noch ein Benutzername, der noch mit übergeben wird (oder bei AUTH_TYPE=ident sameuser.)

ANMERKUNG: AUTH_ARGUMENT enthält nur aus drucktechnischen Gründen eine Zeilenschaltung

Einige Beispiele:						
#	TYP	DATENBANK	BENUTZER	IP-ADRESSE	IP_MASKE	METHODE
local	all		all			trust
Jeder lokale Benutzer kann sich auf dem lokalen System (über Unix-Domain-Sockets) auf jeder Datenbank unter jedem Namen anmelden.						
#	TYP	DATENBANK	BENUTZER	IP-ADRESSE	IP_MASKE	METHODE
host	versuch		all	60.1.2.0	255.255.255.0	trust
Anmeldung auf der Datenbank versuch von jeder Host aus dem Netz 60.1.2.x. Der Benutzername ist der Anmeldename vom Betriebssystem.						
#	TYP	DATENBANK	BENUTZER	IP-ADRESSE	IP_MASKE	METHODE
host	versuch		stefan	60.1.2.3	255.255.255.255	md5
Der Benutzer Stefan meldet sich von der Host 60.1.2.3 auf der Datenbank Versuch an, wenn er das richtige Passwort hat.						
#	TYP	DATENBANK	BENUTZER	IP-ADRESSE	IP_MASKE	METHODE
host	versuch		all	60.0.0.0	255.0.0.0	ident sameuser
Alle Anwender aus dem Netz 60.x.x.x können sich auf der Datenbank versuch anmelden. Der Anmeldename ist der Name den Ident für die Verbindung ermittelt hat.						
#	TYP	DATENBANK	BENUTZER	IP-ADRESSE	IP_MASKE	METHODE
host	versuch		all	60.0.0.0	255.0.0.0	ident abc
Alle Anwender aus dem Netz 60.x.x.x können sich auf der Datenbank versuch anmelden, wenn sie die Ident-Prüfung bestehen. Näheres entnehmen Sie der Beschreibung der Datei pg_ident.conf						

TIPP: Um bei Änderungen im laufenden Betrieb die Datenbank nicht herunter- und wieder hochzufahren, geben Sie den Befehl **pg_ctl reload [-D DATADIR]** ein.

Die Konfigurationsdatei postgres.conf

Auch in dieser Datei befinden sich wichtige Informationen für die Datenbank. Sie legen dort z.B. Verbindungsinformationen (Port, maximale Anzahl von Usern), Verzeichnis von weiteren Konfigurationsdateien und internen Einstellungen (shared Memory usw.) fest. Bei der Windows-Version bearbeiten Sie diese Zeile, damit die anderen PCs auf die Datenbank zugreifen können:

```
#listen_addresses='localhost'
```

Entfernen Sie das Kommentarzeichen und ersetzen Sie localhost gegen einen Stern. Dadurch kann jeder auf die Datenbank zugreifen (siehe unten).

```
listen_addresses='*'
```

Weitere wichtige Punkte sind zum Beispiel die Port-Nr (Standardwert: port = 5432) und maximale Anzahl von Benutzern (Standardwert: max_connections = 100).

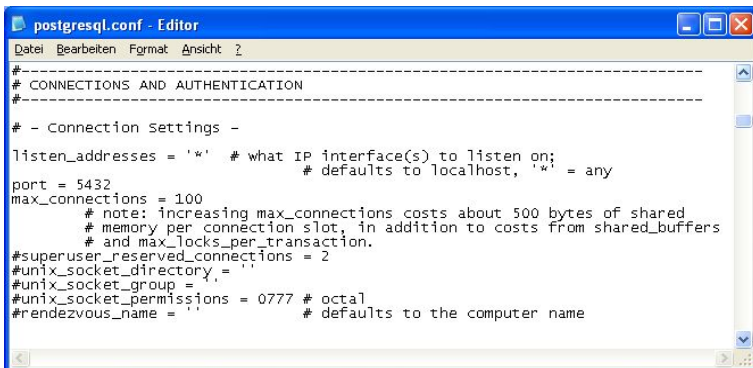


Abbildung 20 postgresql.conf - Inhalt

Die Konfigurationsdatei pg_ident.conf

Diese Datei ist für das Ident-Authentifizierungssystem wichtig. Hier prüft die Datenbank den Benutzernamen des Betriebssystems und ermittelt dann die erlaubten Datenbankbenutzernamen. In dieser Datei tragen Sie neben den Anmeldenamen auch „Map-Namen“. In der Konfigurationsdatei pg_hba.conf tragen Sie zu diesen „Map-Namen“ die Datenbanken ein (**Hinweis:** Der Dämon identd muss auf dem System laufen). Ein Beispiel:

Datei pg_hba.conf

```
# TYP  DATENBANK  BENUTZER  IP-ADRESSE  IP_MASKE  METHODE
host   versuch   all       60.0.0.0    255.0.0.0  ident abc
```

Zu dem „Map-Namen“ abc finden Sie in der Datei pg_ident.conf die entsprechenden Benutzer. An der Datenbank können sich alle Leute anmelden, die unter dem Begriff abc eingetragen sind.

Datei pg_ident.conf

Unter dem Eintrag Ident-Benutzer führen Sie die Anwender mit ihren UNIX-Namen. Unter der Rubrik PG-Benutzer finden Sie die erlaubten Benutzernamen der UNIX-User für die Datenbank. Für einen Unix-Benutzer sind auch mehrere PG-Benutzernamen möglich.

```
# MAPNAME      IDENT-BENUTZER      PG-BENUTZER
abc            stefank             stefan
abc            janina              janina
```

Das Programm postmaster

Das Programm ist das Herz der Datenbank. Läuft es nicht, so kann keiner auf die Daten zugreifen. Das Programm startet einen separaten Serverprozess und überwacht die Verbindung. Weiterhin überwacht es auch die Kommunikation zwischen den Serverprozessen. In der Grundeinstellung läuft das Programm im Vordergrund und gibt verschiedene Meldungen in die Standard-Ausgabe aus. In der Praxis läuft das Programm aber im Hintergrund. Ein gestartetes Programm Postmaster verwaltet nur einen Datenbankcluster. Sie können aber mehrere Postmaster-Programme gleichzeitig laufen lassen. Das Programm können Sie entweder als User **postgres** starten, oder auch als ein anderer PostgreSQL-Benutzer.

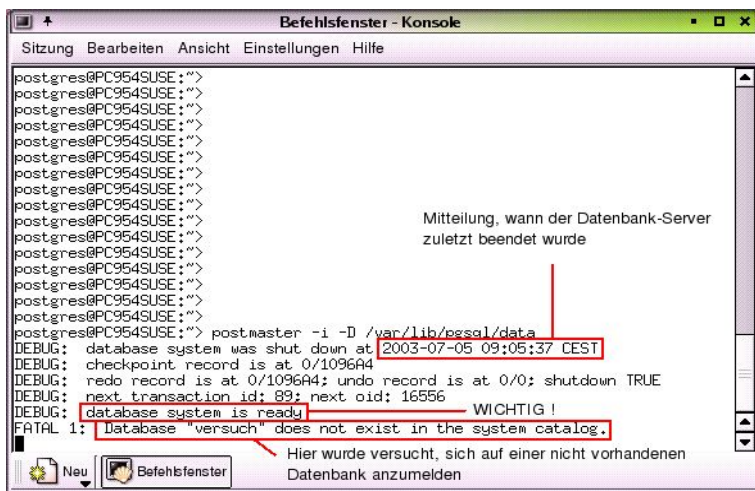


Abbildung 21 Postmaster

Die Parameter zum Programm postmaster

Parameter	Pflicht	Inhalt
-A 0 1	nein	Hilfe zur Fehlersuche, ist nur verfügbar, wenn es bei Compilerlauf eingestellt wurde. In der Grundeinstellung ist es an.
-B [NBUFFERS]	nein	Enthält die Anzahl der shared Buffers. Die Grundeinstellung ist 64
-c [NAME]=[VALUE]	nein	Setzen eines Runtime-Parameters
-d [1-5]	nein	Die verschiedenen Debug Levels. Sie können Werte von 1 bis 5 einstellen. Je höher der Debug-Level ist, umso umfangreichere Meldungen gibt das Programm aus.
-D [DATADIR]	ja	Dieser Parameter enthält das Database Verzeichnis.
-F	nein	Den fsync ausschalten
-h [HOSTNAME]	nein	Der Hostname oder die IP-Adresse auf der die Datenbank läuft
-i	nein	Er ermöglicht eine TCP/IP Verbindung (wichtig bei ODBC!) auf die Datenbank. Wenn mehrere Anwender auf die Datenbank über ODBC zugreifen, ist der Parameter Pflicht.
-k [DIRECTORY]	nein	
-l	nein	Er ermöglicht SSL Verbindungen auf die Datenbank
-N [max. Verb.]	nein	Er steht für die maximale Anzahl von erlaubten Verbindungen (default 32)
-o [OPTIONS]	nein	Er gibt OPTIONEN an den Datenbank-Server weiter
-p [PORT]	nein	Der offenen Port für die Verbindungsaufnahme mit der Datenbank. Per Default ist der Port 5432 vorgesehen.
-S	nein	Startet die Datenbank ohne Ausgabe von Meldungen. Das Programm läuft dann im Hintergrund

Zusätzlich gibt es noch 2 Parameter für die Entwickler der Datenbank.

Parameter	Pflicht	Inhalt
-n	nein	Das shared Memory wird nach einem Absturz nicht reinitialisiert.
-s	nein	Sendet das Signal SIGSTOP an alle Backend-Server, wenn einer abstürzt

Umgebungsvariablen

PGCLIENTENCODING

PGDATA (und wenn weitere vergeben wurden, z.B. PGDATA2)

PGDATASTYLE

PGPORT

TZ

Arbeiten mit der Datenbank

Datenbanken anlegen

Es gibt zwei verschiedene Möglichkeiten eine Datenbank anzulegen (von Haus aus haben Sie die Datenbank template1):

1. Mit Hilfe eines SQL-Befehles (siehe auch Anhang (CREATE DATABASE Datenbankname OWNER Eigentümer))
2. Über das Programm createdb. Die genaue Syntax sieht wie folgt aus:
createdb [-O Owner] Datenbankname
Die Option -O steht für die Zuweisung des neuen Eigentümers (siehe Bild unten).



```
postgres@linux:/home/stefan - Befehlsfenster - Konsole
Sitzung Bearbeiten Ansicht Lesezeichen Einstellungen Hilfe
postgres@linux:/home/stefan> createdb -O stefan versuch
CREATE DATABASE
postgres@linux:/home/stefan>
postgres@linux:/home/stefan>
```

Abbildung 22 Datenbanken anlegen - createdb

Das Programm createdb

Wie schon oben aufgeführt, legen Sie mit diesem Programm neue Datenbanken an. Die Syntax sieht so aus:

createdb [OPTIONEN] [DBNAME] [DESCRIPTION]

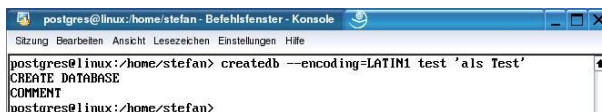
Ihnen stehen folgende Optionen zur Verfügung (**ANMERKUNG:** Die Kurzform der Optionen steht in eckigen Klammern. Wenn Sie mehrere Optionen einsetzen möchten, dann trennen Sie die Optionen durch ein Leerzeichen).

Option	Inhalt
--location=PATH [-D]	Das alternative Verzeichnis für die Datenbank
--encoding=ENCODING [-E]	Der Zeichensatz der Datenbank (Stichwort Umlaute)

Option	Inhalt
--owner=OWNER [-O] (großes o)	Der Eigentümer der Datenbank.
--template=TEMPLATE [-T]	Die Template-Datenbank zum Kopieren.
--echo [-e]	Hier sehen Sie die SQL-Befehle, mit denen die Datenbank angelegt wird
--quiet [-q]	Es wird keine Meldung ausgegeben
--help	Die Hilfe für das Programm

Auch für die Verbindung zu anderen Datenbankservern gibt es noch Optionen:

Option	Inhalt
--host=HOSTNAME [-h]	Der Name der Host
--port=PORT [-p]	Die Port-Nr. der Datenbank
--username=USERNAME [-U] (großes U)	Der Benutzername, mit dem Sie sich auf dem Datenbankserver anmelden
--password [-W] (großes W)	Das Passwort zu dem Benutzer, mit dem Sie sich auf dem Datenbankserver anmelden



```

postgres@linux:/home/stefan - Befehlsfenster - Konsole
Sitzung Bearbeiten Ansicht Lesezeichen Einstellungen Hilfe
postgres@linux:/home/stefan> createdb --encoding=LATIN1 test 'als Test'
CREATE DATABASE
COMMENT
postgres@linux:/home/stefan>
  
```

Abbildung 23 createdb - ein Beispiel

Sollten Sie eventuell Probleme mit dem Plattenplatz haben, so können Sie die neue Datenbank auch in einem anderen Verzeichnis ablegen. Gehen Sie dabei wie unten gezeigt vor:

Schritt	Inhalt
1	Weisen Sie der Variablen PGDATA2 ein Verzeichnis zu (z.B. PGDATA2=/opt/datenbank/daten). Halten Sie diese Information auch in anderen Dateien fest (z.B. /etc/profile oder im Start-Skript). Den Namen der Variablen können Sie übrigens Beliebig vergeben. Wenn PGDATA2 schon benutzt wird, dann vergeben Sie die Variable PGDATA3.
2	Exportieren Sie die Information mit: export PGDATA2
3	Danach richten Sie den Speicherplatz mit dem Befehl initlocation PGDATA2 ein (kein \$ vor dem PGDATA2)
4	Führen Sie einen Restart des Datenbank-Servers durch.
5	Danach legen Sie die Datenbank an. Der Befehl sieht so aus: CREATE DATABASE [name] WITH LOCATION = 'PGDATA2' Das Programm createdb hat für diesen Fall die Option -D .

Datenbanken löschen

Wie auch beim Anlegen einer Datenbank haben Sie zwei Möglichkeiten dazu:

1. Mit Hilfe eines SQL-Befehles (siehe auch Anhang (DROP DATABASE Datenbankname))
2. Über das Programm dropdb. Die Syntax lautet so: **dropdb Datenbankname**

Backup der Datenbank

Eine Sicherung der Datenbank gehört zu den elementarsten Aufgaben des Systemverwalters. Dazu haben Sie die Programme **pg_dump** und **pg_dumpall**. Beide rufen Sie über die Konsole auf. Für einzelne Tabellen können Sie auch den SQL-Befehl copy einsetzen.

Das Programm pg_dump

Mit dem Programm sichern Sie eine Ihrer Datenbanken. Die Ausgabe erfolgt in eine Skriptdatei (enthält Text und SQL-Befehle um die Datenbank so wiederherzustellen, wie sie zum Zeitpunkt der Sicherung war) oder eine andere Archivdatei. Das Programm blockiert keine anderen Benutzer und erstellt konsistente Sicherungsdateien. Während der Sicherung sollten Sie unbedingt die Meldungen lesen. Geben Sie bei dem Befehl keinen Datenbanknamen an, dann nimmt das Programm den Wert aus der Umgebungsvariable PGDATABASE. Ist auch der Wert leer, dann nimmt das Programm die aktuelle Datenbank.

Aufbau:

pg_dump [OPTIONEN] Datenbank

Ihnen stehen folgende Optionen zur Verfügung (**ANMERKUNG:** Die Kurzform der Optionen steht in eckigen Klammern. Wenn Sie mehrere Optionen einsetzen möchten, dann trennen Sie die Optionen durch ein Leerzeichen).

Option	Bedeutung
<code>--data-only</code> [-a]	Sichert nur die Daten, nicht das Schema (die Datendefinitionen)!
<code>--blobs</code> [-b]	Die Lage Objekte werden auch gesichert
<code>--clean</code> [-c] (kleines c)	Zuerst werden die Datenbankobjekte gelöscht und dann erzeugt. Diese Option hilft Ihnen bei reinem Textformat. Bei den anderen Formaten können Sie diese Option angeben, wenn Sie pg_restore ausführen.
<code>--create</code> [-C] (großes C)	Als Erstes legt das Programm die Datenbank an und meldet sich dann bei ihr an. Diese Möglichkeit hilft Ihnen nur bei der Sicherung von Textdateien. Bei den anderen Formaten können Sie diese Option angeben, wenn Sie pg_restore ausführen.
<code>--inserts</code> [-d]	Hier sichern Sie die Daten als INSERT-Befehle und nicht als COPY. Leider läuft mit dieser Einstellung die Wiederherstellung langsamer. Das Archiv lässt sich aber besser auf andere Datenbankprodukte portieren.

Option	Bedeutung
<code>--column-inserts</code> <code>--attribute-inserts</code> <code>[-D]</code>	Die Ausgabe erfolgt in Form von INSERT-Befehlen mit vorgegebenen Spaltennamen (INSERT INTO Tabelle (Spalte, ...) VALUES ...). Leider wirkt sich das negativ auf die Geschwindigkeit des Restores aus.
<code>--file</code> <code>[-f]</code>	Die Ausgabe erfolgt in eine vorgegebene Datei. Ohne diese Angabe wird die Standardausgabe verwendet.
<code>--format</code> <code>[-F]</code>	Hier können Sie das Ausgabeformat der Sicherung wählen: p = die Daten als Text und SQL-Anweisungen für die Struktur t = Ausgabe als ein tar-Archiv. Kann später als Eingabedatei für pg_restore verwendet werden. Schemaelemente können umgeordnet oder auch ausgelassen werden. Sie können auch eingrenzen, welche Daten bei der Wiederherstellung geladen werden sollen. c = Erzeugt ein „Custom“ Archiv, kann als Eingabe für pg_restore verwendet werden. Diese Möglichkeit ist am flexibelsten, da man Daten und Schemaelemente umsortieren kann. Außerdem ist in der Grundeinstellung eine Komprimierung der Informationen vorgesehen.
<code>--ignore-version</code> <code>[-i]</code>	Die Version zwischen dem Programm und dem Datenbankserver wird ignoriert (wichtig: bei früheren Versionen). ACHTUNG: Wenn es nicht funktioniert, dann sagen Sie nicht, Sie seien nicht gewarnt worden.
<code>-n namespace</code>	
<code>--oids</code> <code>[-o]</code>	Gibt auch den Objekt Identifier jeder Tabelle (OIDs) aus. Das ist wichtig, wenn Ihre Anwendung diese Informationen benötigt (siehe Fremdschlüssel, Constraints).
<code>--no-owner</code> <code>[-O]</code> (großes O)	Gibt keine Befehle aus, die dafür sorgen, dass die Objekteigentümer mit der ursprünglichen Datenbank übereinstimmen. Diese Option unterbindet aber nicht alle Verbindungsversuche mit der Datenbank, sondern nur solche, die dazu dienen die Eigentumsverhältnisse wiederherzustellen. Diese Option hilft Ihnen nur beim reinen Textformat. Bei den anderen Formaten können Sie diese Optionen angeben, wenn Sie pg_restore ausführen.

Option	Bedeutung
--no-reconnect [-R]	überholt (Version 7.4.6)
--schema	Sichert nur den Inhalt des Schemas
--schema-only [-s]	Sichert nur das Schema (Definitionen), nicht die Daten.
--superuser=Username [-S Username]	Eingabe des Superusernamens um den Trigger abzuschalten. Das ist nur dann wichtig, wenn --disable-triggers eingesetzt wird.
--table=Tabelle [-t]	Sichert nur eine einzelne Tabelle. Haben Sie in verschiedenen Schemas Tabellen mit dem gleichen Namen, dann werden alle gesichert. Möchten Sie nur eine bestimmte Tabelle sichern, dann nehmen Sie die Parameter --schema und --table. ACHTUNG: Zurzeit wird für diese Option keine Garantie übernommen.
--verbose [-v]	Die Fortschrittsmeldungen werden in die Standardfehlerausgabe geschrieben.
--no-privileges --no-acl [-x]	Verhindert die Sicherung der Zugriffsberechtigungen (siehe grant/revoke Anweisungen).
-X use-set-session-authorization	überholt in der Version 7.4.6
--disable-triggers [-X disable-triggers]	Diese Option ist dann relevant, wenn Sie nur die Daten sichern. Sollten Sie die Daten wieder zurück laden, dann greifen die Prüfungen und Trigger nicht. Diese Option müssen Sie als Superuser ausführen.
--compress=0..9 [-Z]	Hier stellen Sie die Stufe der Komprimierung ein. Zurzeit unterstützt nur das „Custom“ Archivformat diese Option.
--host=HOSTNAME [-h]	Der Name der Host
--port=PORT [-p]	Die Port-Nr. der Datenbank
--username=USERNAME [-U] (großes U)	Der Benutzername, mit dem Sie sich auf dem Datenbankserver anmelden
--password [-W] (großes W)	Das Passwort zu dem Benutzer, mit dem Sie sich auf dem Datenbankserver anmelden

Umgebungsvariablen:

PGDATABASE
PGHOST
PGPORT
PGUSER

Beispiele:

Eine einfach Sicherung in eine Datei:

pg_dump [Datenbank] > [der Dateiname, in die Sie die Sicherung umleiten]

Große Datenbanken können Sie auch komprimieren. Leiten Sie die Ausgabe an das Programm gzip weiter. In der Praxis sieht es so aus:

pg_dump [Datenbank] | gzip > [Name der Datei.gz]

Wenn Sie alle Datenbanken sichern möchten, dann nehmen Sie das Programm ***pg_dumpall***.

Das Programm pg_dumpall

Mit diesem Programm sichern Sie einen kompletten Datenbankcluster. Im Falle eines Restores nehmen Sie das Programm `psql`. Im Gegensatz zum Programm `pg_dump` sichern Sie hier auch globale Objekte! Leider gibt es hier auch eine Einschränkung: „large Objects“ werden leider nicht gesichert (Stand Version 7.4.6). Setzen Sie hier das Programm `pg_dump` ein. Das Programm meldet sich mehrere Male am Datenbankserver an. Damit Sie das Passwort nicht jedesmal eingeben müssen, hinterlegen Sie es in der Datei `$HOME/.pgpass`.

Aufbau:

`pg_dumpall` [OPTIONEN]

Ihnen stehen folgende Optionen zur Verfügung (**ANMERKUNG:** Die Kurzform der Optionen steht in eckigen Klammern. Wenn Sie mehrere Optionen einsetzen möchten, dann trennen Sie die Optionen durch ein Leerzeichen).

<i>Option</i>	<i>Bedeutung</i>
<code>--data-only</code> [-a]	Sichert nur die Daten, nicht das Schema (die Datendefinitionen)!
<code>--clean</code> [-c] (kleines c)	Zuerst werden die Datenbankobjekte gelöscht und dann erzeugt. Diese Option hilft Ihnen bei reinem Textformat. Bei den anderen Formaten können Sie diese Option angeben, wenn Sie <code>pg_restore</code> ausführen.
<code>--inserts</code> [-d]	Hier sichern Sie die Daten als INSERT-Befehle und nicht als COPY. Leider läuft mit dieser Einstellung die Wiederherstellung langsamer. Das Archiv lässt sich aber besser auf andere Datenbankprodukte portieren.
<code>--column-inserts</code> <code>--attribute-inserts</code> [-D]	Die Ausgabe erfolgt in Form von INSERT-Befehlen mit vorgegebenen Spaltennamen (INSERT INTO Tabelle (Spalte, ...) VALUES ...). Leider wirkt sich das negativ auf die Geschwindigkeit des Restores aus.
<code>--globals-only</code> [-g]	Nur die globalen Objekte werden gesichert (Anwender und Gruppen), keine Datenbanken.

Option	Bedeutung
<code>--ignore-version</code> [<code>-i</code>]	Die Version zwischen dem Programm und dem Datenbankserver wird ignoriert (wichtig: bei früheren Versionen). ACHTUNG: Wenn es nicht funktioniert, dann sagen Sie nicht Sie seien nicht gewarnt worden).
<code>--oids</code> [<code>-o</code>]	Gibt auch den Objekt Identifier jeder Tabelle (OIDs) aus. Das ist wichtig, wenn Ihre Anwendung diese Informationen benötigt (siehe Fremdschlüssel, Constraints).
<code>--schema-only</code> [<code>-s</code>]	Sichert nur das Schema und keine Daten.
<code>--verbose</code> [<code>-v</code>]	Die Fortschrittsmeldungen werden in die Standardfehlerausgabe geschrieben.
<code>--no-privileges</code> <code>--no-acl</code> [<code>-x</code>]	Verhindert die Sicherung der Zugriffsberechtigungen (siehe <code>grant/revoke</code> Anweisungen).
<code>--host=HOSTNAME</code> [<code>-h</code>]	Der Name der Host
<code>--port=PORT</code> [<code>-p</code>]	Die Port-Nr. der Datenbank
<code>--username=USERNAME</code> [<code>-U</code>] (großes U)	Der Benutzername, mit dem Sie sich auf dem Datenbankserver anmelden
<code>--password</code> [<code>-W</code>] (großes W)	Das Passwort zu dem Benutzer, mit dem Sie sich auf dem Datenbankserver anmelden

Umgebungsvariablen:

PGHOST
PGPORT
PGUSER

Beispiel:

alle Datenbanken sichern:

pg_dumpall > sicher.out

Um diese Datenbank wiederherzustellen (die Datenbank, an der Sie sich

gerade angemeldet haben, ist unerheblich, die Skriptdatei enthält alle Befehle, um die gesicherte Datenbank zu erzeugen und sich dann mit ihr zu verbinden):

psql -f sicher.out template1

Restore der Datenbank

Für die Wiederherstellung stehen Ihnen 2 Möglichkeiten zur Verfügung:

1. Das Programm psql
2. Das Programm pg_restore

Mit dem Programm psql

Mit dem psql-Befehl können Sie Ihre Daten aus der Sicherungskopie wieder in die Datenbank zurück spielen. Die Syntax siehe wie folgt aus:

psql -d [Datenbank] < [der Dateiname Ihrer Sicherungsdatei]

Bitte beachten Sie, dass das psql-Kommando keine Datenbank neu anlegt! Komprimierte Dateien stellen Sie so wieder her (ggf. muss die Datenbank vorher noch angelegt werden):

cat [Name der Sicherungsdatei.gz] | gunzip | psql [Name der Datenbank]

Anmerkung: Über den Weg Backup und Restore können Sie auch Daten von einem Server zu einem anderen übertragen. Der Befehl sieht dann so aus:

pg_dump -h [Host1] [Datenbank] | psql -h [Host2] [Datenbank]

Das Programm pg_restore

Mit diesem Programm stellen Sie eine Datenbank wieder her. Die Ausgangsdatei muss vom Programm pg_dump erstellt worden sein. Wenn Sie keinen Dateinamen angeben, dann nimmt das Programm die Standardeingabe.

Aufbau:

pg_restore [OPTIONEN] [Dateiname]

Ihnen stehen folgende Optionen zur Verfügung (**ANMERKUNG:** Die Kurzform der Optionen steht in eckigen Klammern. Wenn Sie mehrere Optionen einsetzen möchten, dann trennen Sie die Optionen durch ein Leerzeichen).

Option	Bedeutung
--data-only [-a]	Sichert nur die Daten, nicht das Schema zurück!
--clean [-c] (kleines c)	Zuerst werden die Datenbankobjekte gelöscht und dann erzeugt.
--create [-C]	Legt die Datenbank an, bevor sie zurück kopiert wird. (Wenn Sie diese Option verwenden, dann legt das Programm die in der Option -d übergebene Datenbank nur an und kopiert die Daten aus in Archiv in die Datenbank).
--inserts [-d]	Das Programm verbindet sich mit der Datenbank und überträgt die Daten in diese Datenbank.
--file=dateiname [-f]	Name der Ausgangsdatei
--format=Format [-F]	Das Format der Archivdatei t=Tar-Archiv c="Custom" Format, ggf. komprimiert, Daten und Schemaelemente können umsortiert werden.
--ignore-version [-i]	Die Versionsprüfung wird ignoriert
--index=Index [-I] (großes i)	Der in der Definition benannte Index wird wieder hergestellt.

<i>Option</i>	<i>Bedeutung</i>
--list [-l] (kleines l)	Gibt den Inhalt des Archivs wieder.
--use-list=Datei [-L Datei]	Nur die in der Datei aufgeführten Objekte werden wiederhergestellt. Durch ein Semikolon am Anfang können Sie Zeilen auskommentieren.
--orig-order [-N]	Das Programm stellt die Objekte in der Reihenfolge her, in der man sie angelegt hat.
--host=HOSTNAME [-h]	Der Name der Host
--port=PORT [-p]	Die Port-Nr. der Datenbank
-- username=USERNAME [-U] (großes U)	Der Benutzername, mit dem Sie sich auf dem Datenbankserver anmelden
--password [-W] (großes W)	Das Passwort zu dem Benutzer, mit dem Sie sich auf dem Datenbankserver anmelden

Plattenplatz optimieren

Bei den Update- oder Delete-Befehlen werden unter Umständen nicht sofort die alten Daten entfernt. Mit dem SQL-Befehl VACUUM bereinigen Sie den Plattenplatz. Sie können ihn entweder gezielt bei einer Tabelle anwenden, oder auch auf die ganze Datenbank. Seit der Version 7.2 können Sie den VACUUM-Befehl neben den normalen Operationen wie Select, Insert, Update und Delete einsetzen. Wenn Sie etwas an einer Tabellenstruktur ändern, darf er nicht laufen! Setzen Sie ihn am Besten in den Zeiten geringer Aktivitäten ein. Verwenden Sie zur Komprimierung nur den VACUUM-Befehl, nicht den VACUUM FULL-Befehl. Den VACUUM FULL-Befehl benötigen Sie, wenn Sie die meisten Einträge in einer Tabelle gelöscht haben. Überprüfen Sie auch immer den Verbrauch an Plattenplatz. (z.B. mit dem Befehl **contrib/dbsize** aus dem Programm psq!). Siehe auch Programm Vacuumdb.

Routine Reindexing

Die Datenbank kann leider die B-tree Indexseiten nicht richtig wieder verwenden. Deshalb müssen Sie von Zeit zu Zeit die Seiten wieder neu aufbauen. Diese Aktionen erledigen Sie mit dem Befehl **REINDEX**. **ACHTUNG:** Wenn Sie den Befehl bei Systemtabellen und -Indexen anwenden möchten, dann starten Sie das Postgres im Single-Use Mode (siehe Programm postgres)!

Migration der Datenbank / Einführung einer neuen Version

Bei einem größeren Versionswechsel müssen Sie die Datenbank neu aufbauen. Als Faustformel gilt hierbei: Ändert sich die Zahl nach dem ersten Punkt in der Versionsnummer, so ist die Datenbank mitbetroffen (z.B. von 7.3 auf 7.4). Bei einem Versionswechsel nach dem zweiten Punkt tauschen Sie nur die Programme aus (z.B. von 7.3.3 auf 7.3.4).

Bei einem Releasewechsel gehen Sie mit der Installation der Programme so vor, wie im Kapitel **Installation einer neuen Version aus dem Internet** beschrieben (ändern Sie aber das Zielverzeichnis ab, nicht unbedingt /usr/local/postgresql). Danach pflegen Sie noch die entsprechenden Variablen. Sie können übrigens auch ohne Probleme zwei unterschiedliche Versionen auf einem Host laufen lassen. Sie müssen nur mit verschiedenen Portnummern arbeiten.

Wenn Sie zwei Versionen auf einem Host laufen lassen, dann können Sie die Datenbank in einem Zug umstellen:

pg_dumpall -p 5432 | psql -d template1 -p 6543

Eine weitere Möglichkeit der Migration der Daten sieht so aus:

Schritt	Aktion
1	Wechseln Sie in das Home-Verzeichnis vom User postgres (z.B. /var/lib/postgresql)
2	Führen Sie eine Datensicherung durch: <i>pg_dumpall > backup</i>
3	Fahren Sie danach die Datenbank herunter: <i>pg_ctl stop -D /var/lib/postgresql/data</i>
4	Jetzt sollten Sie sich nochmals vergewissern, dass Sie bei den nächsten Schritten nicht versehentlich die alten Programmversionen aufrufen.
5	Legen Sie ein neues Data-Verzeichnis an: z.B. /var/lib/postgresql/data740
6	Mit dem Befehl <i>initdb -D /var/lib/postgresql/data740</i> richten Sie nun die neue Datenbank ein.
7	Bearbeiten Sie die Einstellungen in den Konfigurationsdateien (pg_hba.conf und postgresql.conf).
8	Starten Sie nun die Datenbank (z.B. mit dem Programm <i>postmaster -D /var/lib/postgresql/data740</i>)

Schritt	Aktion
9	Im letzten Schritt spielen Sie das Backup vom Anfang wieder ein: <i>psql template1 < backup</i>

Datenbank Aktivitäten überwachen

Als Administrator haben Sie dazu verschiedene Möglichkeiten:

1. Sie lassen den Postmaster-Prozess im Vordergrund laufen und können so die wichtigsten Meldungen sehen.
2. Eine Log-Datei protokolliert alle wichtigen Ereignisse mit (**Anmerkung:** Mit dem tail-Befehl können Sie sich die letzten Zeilen der Datei ansehen).
3. Mit dem ps - Kommando sehen Sie auch alle Aktivitäten auf dem Host. Die Anweisung sieht so aus:

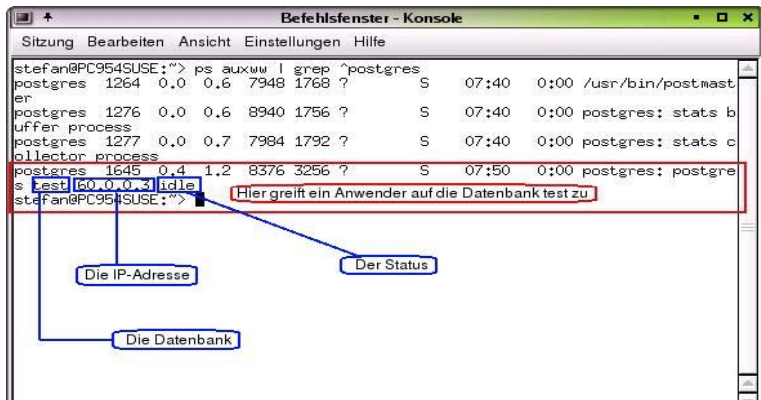


Abbildung 24 Datenbank Aktivitäten überwachen

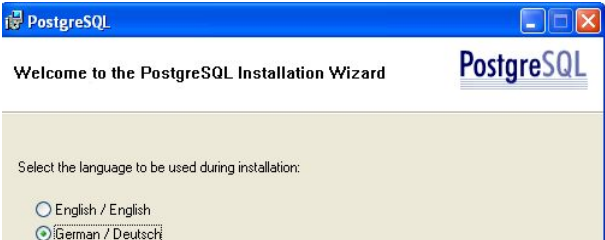
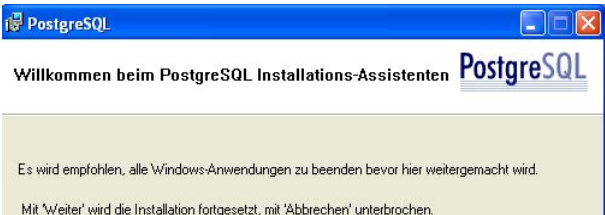
ps auxww | grep ^postgres .


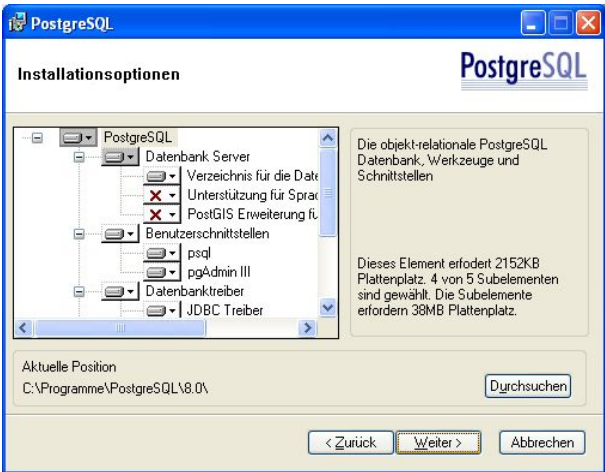
ANMERKUNG: Auch mit dem SQL-Befehl ***select * from pg_stat_activity;*** lassen sich die Aktivitäten anzeigen.

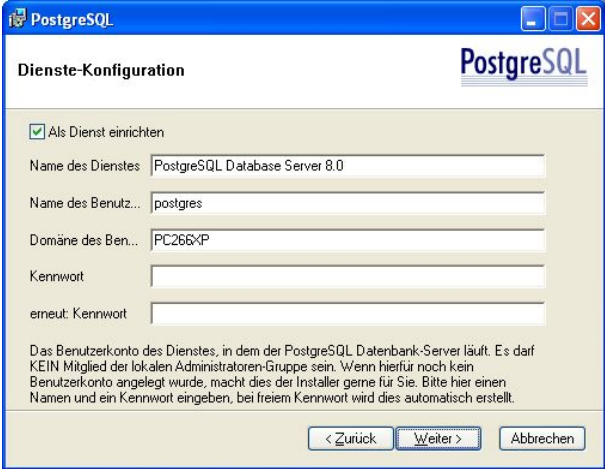
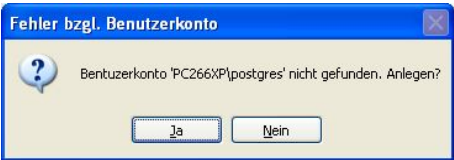
Version 8 – Postgres und Windows


Installation


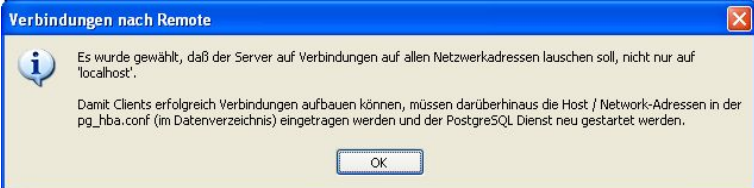
Mit der neuen Version 8.0 ist es nun auch möglich, ohne CYGWIN die Datenbank auf einem Windows-Rechner zu betreiben. Ich habe die neue Version (zurzeit noch als RC1-Version) auf einem Windows 2003 - Server installiert. Laut Homepage (pgFoundry <http://pgfoundry.org/projects/pginstaller>.) lässt sich die Datenbank auf Windows NT/2000/XP und 2003 installieren (für Windows 95/98/Me benötigen Sie immer noch CYGWIN). **WICHTIG:** Als Filesystem ist NTFS Pflicht! Es gibt übrigens eine extra japanische Version. Sie erkennen Sie an dem Zusatz ja.

Schritt	Inhalt / Aktion
1	Anmelden am PC mit Administrationsrechten. Starten Sie den Vorgang mit dem Explorer durch einen Doppelklick auf die MSI-Datei. Nehmen Sie hier die Datei postgresql-8.0.msi .
2	 <p>Abbildung 25 Version 8 Windows, Installation, Sprache auswählen</p> <p>Wählen Sie hier die Sprache aus und drücken den Knopf Start.</p>
3	 <p>Abbildung 26 Version 8 Windows, Installation</p> <p>Auch hier die Informationen durchlesen, der Empfehlung Folge leisten und dann den Button Weiter drücken.</p>

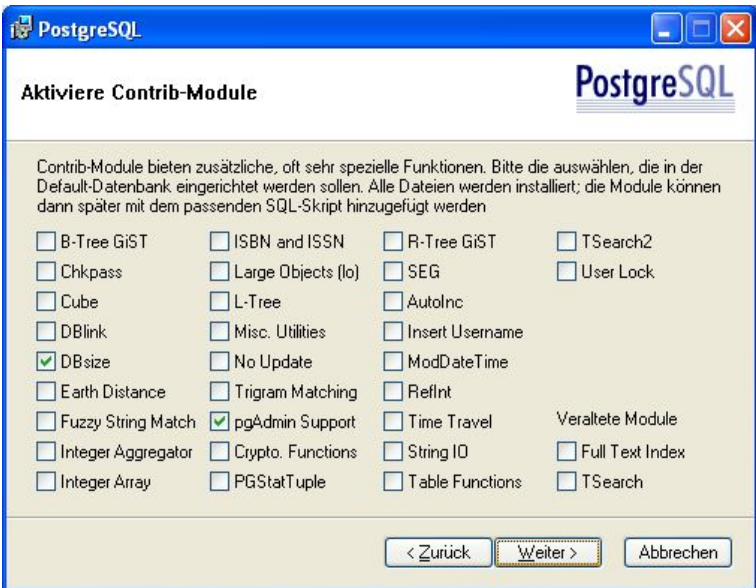
Schritt	Inhalt / Aktion
4	<div data-bbox="344 220 949 694"></div> <p data-bbox="434 694 860 715"><i>Abbildung 27 Version 8 Windows, Installation, Hinweise</i></p> <p data-bbox="260 730 1036 786">Lesen Sie sich die Installationshinweise noch durch und drücken den Knopf Weiter.</p>
5	<div data-bbox="344 799 949 1273"></div> <p data-bbox="481 1273 812 1294"><i>Abbildung 28 Version 8 Windows, Optionen</i></p> <p data-bbox="260 1310 1036 1393">Hier können Sie noch einige Optionen abwählen bzw. verändern. Alle wichtigen Konfigurationsdateien finden Sie dann im Verzeichnis DATA (unterhalb von ..\8.0).</p>


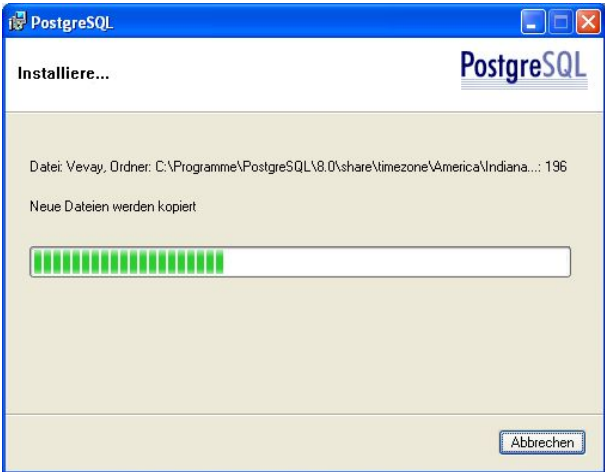
Schritt	Inhalt / Aktion
6	<div data-bbox="344 223 949 694"></div> <p data-bbox="434 695 860 715"><i>Abbildung 29 Version 8 Windows, Dienste-Konfiguration</i></p> <p data-bbox="260 770 1036 983">Wie schon bei der Unix-Version, empfiehlt es sich unter Windows die Datenbank auch als Dienst zu betreiben. Das Installationsprogramm legt Ihnen den Benutzer automatisch neu an. Sie füllen einfach nur die Felder aus (ANMERKUNG: Aus Sicherheitsgründen ohne Adminrechte). Vergeben Sie auch ein Passwort für das Benutzerkonto. Wenn Sie ein einfaches Passwort vergeben, z.B. der Name des Benutzerkontos, dann weist Sie das Installationsprogramm später darauf hin.</p>
7	<div data-bbox="420 997 873 1157"></div> <p data-bbox="464 1158 828 1177"><i>Abbildung 30 Version 8 Windows, User postgres</i></p> <p data-bbox="260 1233 1036 1283">Wenn Sie beim Schritt 6 einen neuen User vorgesehen haben, fragt das Programm noch mal nach.</p>

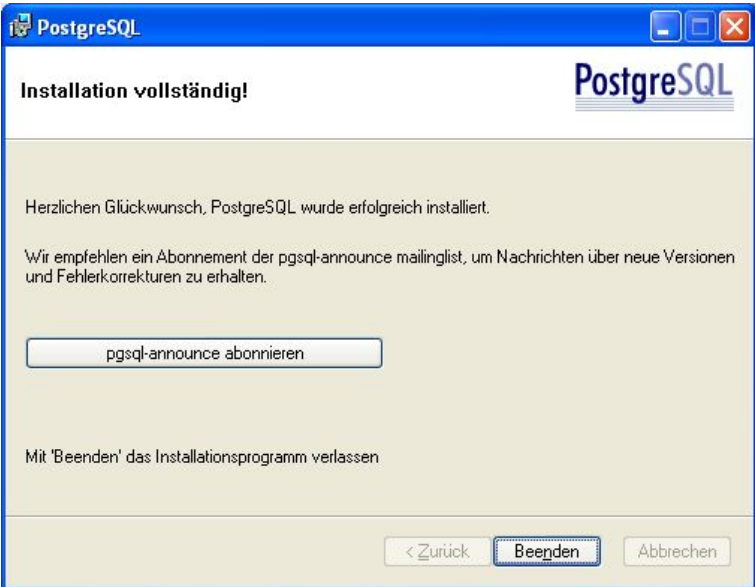
Schritt	Inhalt / Aktion
8	<div data-bbox="269 220 1025 373">  <p>Abbildung 31 Version 8 Windows, Password</p> </div> <p>Ggf. kann es bei der Vergabe des Passworts ein Problem geben (siehe auch Schritt 6, zu kurz, keine Zahlen usw.). Hier können Sie ein zufälliges Passwort vergeben.</p>
9	<div data-bbox="348 539 947 735">  <p>Abbildung 32 Version 8 Windows, Service Rechte</p> </div> <p>Zur Bestätigung, dass alles funktioniert, erhalten Sie diese Meldung.</p>

Schritt	Inhalt / Aktion
10	 <p><i>Abbildung 33 Version 8 Windows, database cluster</i></p> <p>In diesem Schritt legen Sie nun den Superuser der Datenbank an (der User mit den Adminrechten beim Programm psql). Verwechseln Sie diesen User nicht mit dem User postgres im Schritt 6! Stellen Sie noch die Werte für die Felder Locale und Encoding ein. Wichtig ist auch noch das Passwort für den Superuser der Datenbank und auch die Haken im Eintrag Datenbank-Cluster initialisieren und Verbindung auf allen Netzadressen annehmen, nicht nur auf Localhost. Stellen Sie bitte auch noch das Encoding auf LATIN1.</p>
11	 <p><i>Abbildung 34 Version 8 Windows, remote Zugriff</i></p> <p>Bitte den Hinweis beachten und später berücksichtigen.</p>

Schritt	Inhalt / Aktion
12	 <p><i>Abbildung 35 Version 8 Windows, prozedurale Sprachen</i></p> <p>Belassen Sie hier den Vorschlag PL/pgsql. Wenn Sie später Funktionen auf der Datenbank entwickeln möchten, ist die prozedurale Sprache notwendig.</p>

Schritt	Inhalt / Aktion																																				
13	<div><p>Aktiviere Contrib-Module</p><p>Contrib-Module bieten zusätzliche, oft sehr spezielle Funktionen. Bitte die auswählen, die in der Default-Datenbank eingerichtet werden sollen. Alle Dateien werden installiert; die Module können dann später mit dem passenden SQL-Skript hinzugefügt werden</p><table><tr><td><input type="checkbox"/> B-Tree GIST</td><td><input type="checkbox"/> ISBN and ISSN</td><td><input type="checkbox"/> R-Tree GIST</td><td><input type="checkbox"/> TSearch2</td></tr><tr><td><input type="checkbox"/> Chkpass</td><td><input type="checkbox"/> Large Objects (lo)</td><td><input type="checkbox"/> SEG</td><td><input type="checkbox"/> User Lock</td></tr><tr><td><input type="checkbox"/> Cube</td><td><input type="checkbox"/> L-Tree</td><td><input type="checkbox"/> AutoInc</td><td></td></tr><tr><td><input type="checkbox"/> DBlink</td><td><input type="checkbox"/> Misc. Utilities</td><td><input type="checkbox"/> Insert Username</td><td></td></tr><tr><td><input checked="" type="checkbox"/> DBsize</td><td><input type="checkbox"/> No Update</td><td><input type="checkbox"/> ModDateTime</td><td></td></tr><tr><td><input type="checkbox"/> Earth Distance</td><td><input type="checkbox"/> Trigram Matching</td><td><input type="checkbox"/> RefInt</td><td></td></tr><tr><td><input type="checkbox"/> Fuzzy String Match</td><td><input checked="" type="checkbox"/> pgAdmin Support</td><td><input type="checkbox"/> Time Travel</td><td>Veraltete Module</td></tr><tr><td><input type="checkbox"/> Integer Aggregator</td><td><input type="checkbox"/> Crypto. Functions</td><td><input type="checkbox"/> String IO</td><td><input type="checkbox"/> Full Text Index</td></tr><tr><td><input type="checkbox"/> Integer Array</td><td><input type="checkbox"/> PGStatTuple</td><td><input type="checkbox"/> Table Functions</td><td><input type="checkbox"/> TSearch</td></tr></table><p>< Zurück Weiter > Abbrechen</p></div> <p><i>Abbildung 36 Version 8 Windows, weitere Module</i></p> <p>Hier können Sie noch zusätzliche Module aktivieren. Belassen Sie es bei den Vorschlägen und drücken Sie den Knopf Weiter.</p>	<input type="checkbox"/> B-Tree GIST	<input type="checkbox"/> ISBN and ISSN	<input type="checkbox"/> R-Tree GIST	<input type="checkbox"/> TSearch2	<input type="checkbox"/> Chkpass	<input type="checkbox"/> Large Objects (lo)	<input type="checkbox"/> SEG	<input type="checkbox"/> User Lock	<input type="checkbox"/> Cube	<input type="checkbox"/> L-Tree	<input type="checkbox"/> AutoInc		<input type="checkbox"/> DBlink	<input type="checkbox"/> Misc. Utilities	<input type="checkbox"/> Insert Username		<input checked="" type="checkbox"/> DBsize	<input type="checkbox"/> No Update	<input type="checkbox"/> ModDateTime		<input type="checkbox"/> Earth Distance	<input type="checkbox"/> Trigram Matching	<input type="checkbox"/> RefInt		<input type="checkbox"/> Fuzzy String Match	<input checked="" type="checkbox"/> pgAdmin Support	<input type="checkbox"/> Time Travel	Veraltete Module	<input type="checkbox"/> Integer Aggregator	<input type="checkbox"/> Crypto. Functions	<input type="checkbox"/> String IO	<input type="checkbox"/> Full Text Index	<input type="checkbox"/> Integer Array	<input type="checkbox"/> PGStatTuple	<input type="checkbox"/> Table Functions	<input type="checkbox"/> TSearch
<input type="checkbox"/> B-Tree GIST	<input type="checkbox"/> ISBN and ISSN	<input type="checkbox"/> R-Tree GIST	<input type="checkbox"/> TSearch2																																		
<input type="checkbox"/> Chkpass	<input type="checkbox"/> Large Objects (lo)	<input type="checkbox"/> SEG	<input type="checkbox"/> User Lock																																		
<input type="checkbox"/> Cube	<input type="checkbox"/> L-Tree	<input type="checkbox"/> AutoInc																																			
<input type="checkbox"/> DBlink	<input type="checkbox"/> Misc. Utilities	<input type="checkbox"/> Insert Username																																			
<input checked="" type="checkbox"/> DBsize	<input type="checkbox"/> No Update	<input type="checkbox"/> ModDateTime																																			
<input type="checkbox"/> Earth Distance	<input type="checkbox"/> Trigram Matching	<input type="checkbox"/> RefInt																																			
<input type="checkbox"/> Fuzzy String Match	<input checked="" type="checkbox"/> pgAdmin Support	<input type="checkbox"/> Time Travel	Veraltete Module																																		
<input type="checkbox"/> Integer Aggregator	<input type="checkbox"/> Crypto. Functions	<input type="checkbox"/> String IO	<input type="checkbox"/> Full Text Index																																		
<input type="checkbox"/> Integer Array	<input type="checkbox"/> PGStatTuple	<input type="checkbox"/> Table Functions	<input type="checkbox"/> TSearch																																		

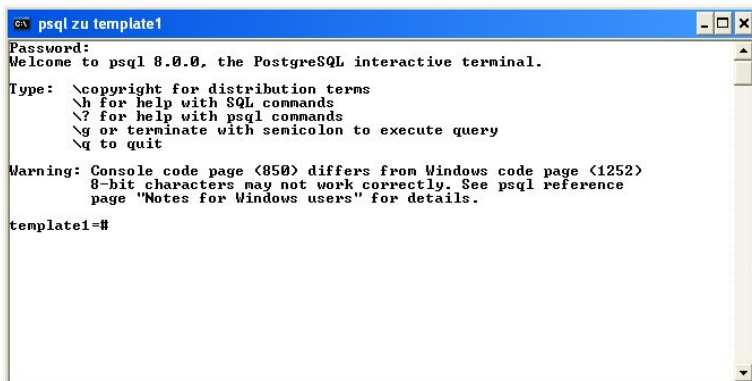
Schritt	Inhalt / Aktion
14	<div></div> <p><i>Abbildung 37 Version 8 Windows, Ende der Installation</i></p> <p>Nun ist alles eingestellt und fertig für die Installation. Drücken Sie die Weiter-Taste.</p>
15	<div></div> <p><i>Abbildung 38 Version 8 Windows, Dateien kopieren</i></p> <p>Danach kopiert das Programm alle notwendigen Dateien.</p>

Schritt	Inhalt / Aktion
16	 <p>Am Ende erhalten Sie diese Mitteilung. Mit dem Druck auf den Knopf Beenden ist die Installation fertig.</p>
17	<p>Damit alle anderen Hosts auf die Datenbank zugreifen können, ändern Sie noch die Zeile in der Datei postgresql.conf ab (über: Start-Taste, Programme, PostgreSQL 8.0, Konfigurationsdateien, postgresql.conf bearbeiten)</p> <pre>#listen_addresses='localhost' in listen_addresses='*'</pre>
18	<p>Die Datei pg_hba.conf noch bearbeiten (über: Start-Taste, Programme, PostgreSQL 8.0, Konfigurationsdateien, pg_hba.conf bearbeiten) (siehe auch Schritt 11).</p>

Die wichtigen Konfigurationsdateien wie postgres.conf und pg_hba.conf finden Sie im \Data-Verzeichnis. Da die Dateien aus einer UNIX-Umgebung kommen, haben Sie in der Beta-Version die Zeilenschaltung von UNIX mit übernommen (Bei der RC1-Version ist das nicht mehr der Fall).

Das Programm *psql* konfigurieren

Wenn Sie das erste Mal das Programm *psql* starten, dann erhalten Sie folgende Meldung:



```
psql zu template1
Password:
Welcome to psql 8.0.0, 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

Warning: Console code page (850) differs from Windows code page (1252)
        8-bit characters may not work correctly. See psql reference
        page "Notes for Windows users" for details.

template1=#
```

Abbildung 39 Version 8 Windows, *psql* Hinweis

Das Problem können Sie auch recht einfach beheben. Rufen Sie den Command-Prompt auf (**Start**-Taste, **Ausführen** und dann **CMD** eingeben). Arbeiten Sie danach die unten gezeigten Schritte ab.

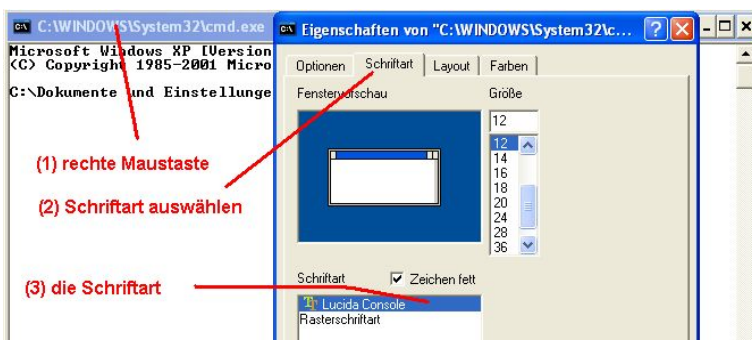


Abbildung 40 Version 8 Windows, CMD

Damit Sie nicht jedes Mal die Einstellung vornehmen müssen, stellen Sie ein, dass die Eigenschaften für jedes Fenster gelten.

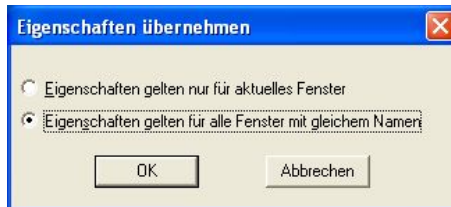
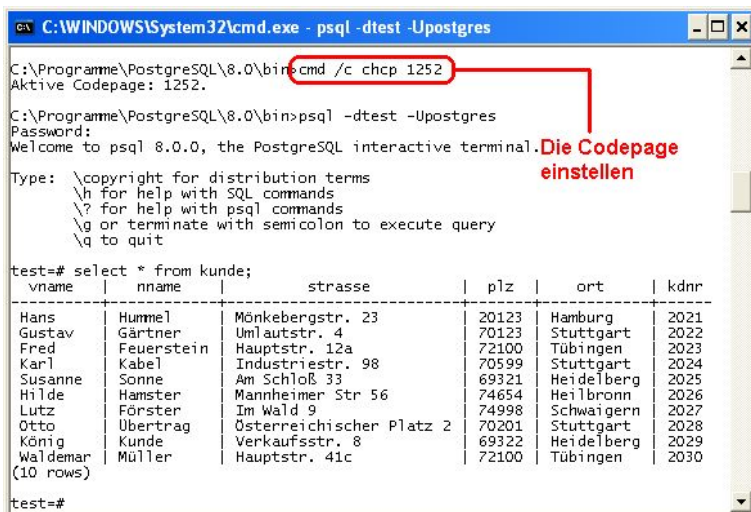


Abbildung 41 Version 8 Windows, CMD Eigenschaften übernehmen

Vor dem Start vom Programm psql ändern Sie noch die Codepage



```
C:\WINDOWS\System32\cmd.exe - psql -dtest -Upostgres
C:\Programme\PostgreSQL\8.0\bin>cmd /c chcp 1252
Aktive Codepage: 1252.

C:\Programme\PostgreSQL\8.0\bin>psql -dtest -Upostgres
Password:
Welcome to psql 8.0.0, 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

test=# select * from kunde;
 vname | nname | strasse | plz | ort | kdnr
-----+-----+-----+----+---+----
Hans   | Hummel | Mönkebergstr. 23 | 20123 | Hamburg | 2021
Gustav | Gärtner | Umlautstr. 4 | 70123 | Stuttgart | 2022
Fred   | Feuerstein | Hauptstr. 12a | 72100 | Tübingen | 2023
Karl   | Kabel | Industriestr. 98 | 70599 | Stuttgart | 2024
Susanne | Sonne | Am Schloß 33 | 69321 | Heidelberg | 2025
Hilde  | Hamster | Mannheimer Str 56 | 74654 | Heilbronn | 2026
Lutz   | Förster | Im Wald 9 | 74998 | Schwaigern | 2027
Otto   | Übertrag | Österreichischer Platz 2 | 70201 | Stuttgart | 2028
König  | Kunde | Verkaufsstr. 8 | 69322 | Heidelberg | 2029
Waldemar | Müller | Hauptstr. 41c | 72100 | Tübingen | 2030
(10 rows)

test=#
```

Abbildung 42 Version 8 Windows, Codepage

Die Angaben können Sie natürlich auch in eine Batch-Datei schreiben.

Arbeiten mit der Datenbank

Den Dienst der Datenbank können Sie recht einfach beenden. Auch das Programm pgAdmin III befindet sich schon auf dem PC. Das Programm **psql** finden Sie unter dem Eintrag **psql to template1**.

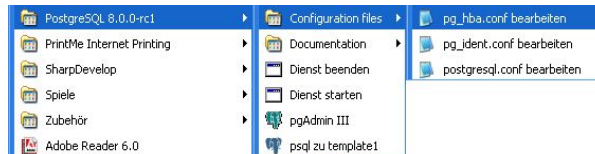


Abbildung 43 Version 8.0 Windows - Arbeiten mit psql

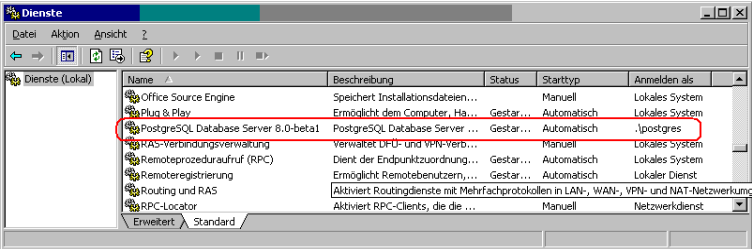
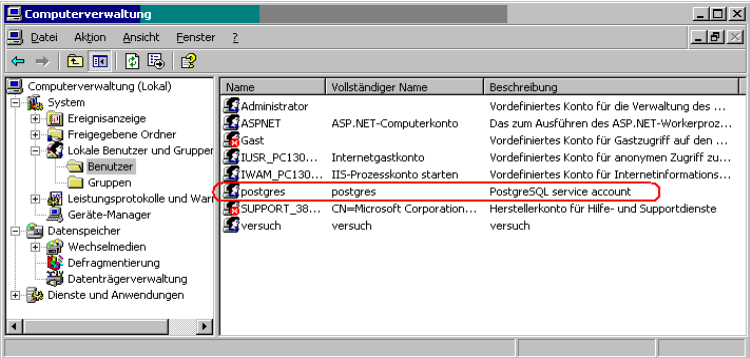
TIPP: Über den NET-Befehl können Sie auch die Datenbank beenden und wieder neu starten.

NET START <<Name des Dienstes, lt. Windows Dienste in der Systemsteuerung (z.B.: pgsq1-8.0-beta2.dev3)>> ==> Datenbank starten

NET STOP <<Name des Dienstes, lt. Windows Dienste in der Systemsteuerung (z.B.: pgsq1-8.0-beta2.dev3)>> ==> Datenbank beenden

ANMERKUNG: In den etwas neueren Versionen können Sie die wichtigsten Konfigurationendateien bequem aus dem Menüsystem aufrufen und mit dem Programm NOTEPAD bearbeiten

Deinstallation

Schritt	Inhalt / Aktion
1	Prüfen Sie, welche Daten für Sie auf der Datenbank noch wichtig sind und sichern Sie die Informationen.
2	<p>Beenden Sie den Dienst der Datenbank (z.B. über die Systemsteuerung, Verwaltung und Dienste)</p> <div></div> <p>Abbildung 44 Version 8 Windows, Dienste</p>
3	<p>Über den Eintrag Software, in der Systemsteuerung, finden Sie die Datenbank (wenn Sie die Software über die MSI-Datei installiert haben). Wählen Sie Postgres aus und drücken Sie deinstallieren.</p>
4	<p>Entfernen Sie noch den User Postgres aus der Benutzerverwaltung des Computers, soweit er angelegt wurde.</p> <div></div> <p>Abbildung 45 Version 8 Windows, Computerverwaltung</p>

Client-Software und andere Zugriffsmöglichkeiten

Nach dem nun die Datenbank läuft, möchten Sie natürlich auch mit verschiedenen Programmen auf die Daten zugreifen. Hier erhalten Sie einen Auszug der Möglichkeiten.

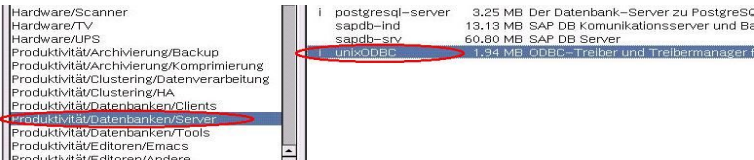
Der Postgres Datenbankserver auf Ihrem LINUX-Host	Schnittstellen	Programme / Möglichkeiten
	unixODBC	Verschiedene Programme, die unixODBC nutzen.
	Windows ODBC	Mögliche Programme wie z.B.: <ul style="list-style-type: none">- ACCESS- OpenOffice- jedes andere Programm, das Datenbanken über ODBC nutzt
	psql	Das Standardprogramm, mit dem Sie über SQL-Befehle alles auf Ihrem Postgres Datenbankserver machen können.
	JDBC (org.postgresql.Driver)	Ihr JAVA-Programm
	libpg (C-Library)	Ihr C-Programm
	Libpqxx (C++-Library)	Ihr C++-Programm
	Npgsql.dll	Ihr C#-Programm

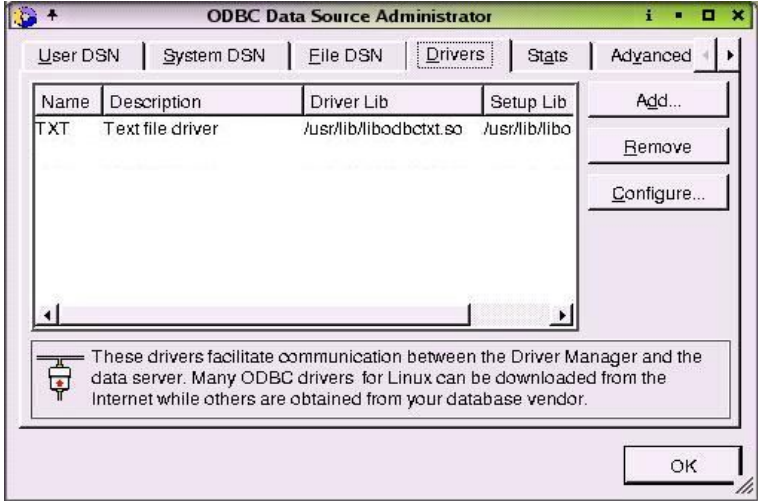
ANMERKUNG: Zurzeit gibt es noch Probleme mit dem unixODBC-Treiber. UnixODBC delegierte die Aufgabe an die Entwicklergruppe von Postgres zurück. Unter der SuSE 8.0 Version hatte ich keine Probleme. Bei der SuSE 8.2-Version fehlten die entsprechenden Dateien. In neueren SuSE-Versionen sind sie aber wieder vorhanden. Bei der Red Hat 9 Distribution fehlen bei unixODBC auch die grafischen Konfigurationsprogramme.


Installation der Programme psql und unixODBC auf einem LINUX-Host

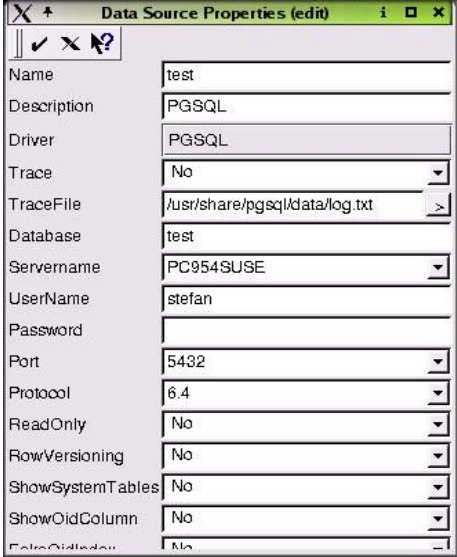
SuSE-Distribution

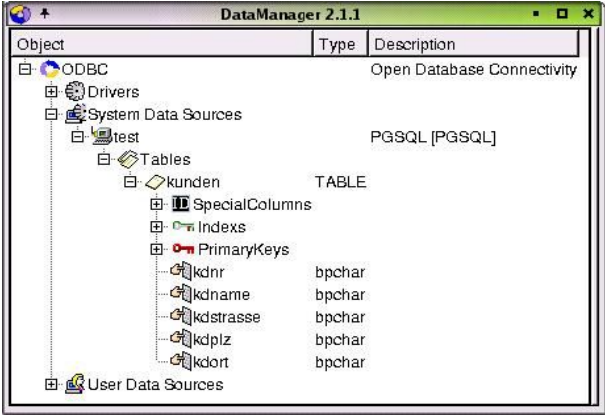
Damit Sie über andere Programme auf Ihre Datenbank zugreifen können, müssen Sie die Client-Software auf Ihrem LINUX-Host installieren. Das Programm **psql** ermöglicht Ihnen nur die Ausführung von SQL-Statements. Im hier gezeigten Beispiel sehen Sie wie Sie unixODBC auf Ihrem LINUX-Host installieren. Das Programm können Sie übrigens auch gleich mit dem Server installieren. Im folgenden Beispiel sehen Sie die Installation unter der SuSE-Distribution. Alternativ können Sie die entsprechenden Programme natürlich auch wie im Kapitel **Installation einer Version aus dem Internet** beschrieben installieren.

Schritt	Inhalt / Aktion															
1	Rufen Sie als erstes das YaST2-Kontrollzentrum auf. Wenn Sie den KDE-Desktop einsetzen, drücken Sie hier die Start-Taste, Einstellungen, Yast menu und dann Yast Kontrollzentrum .															
2	<p>Unter der Rubrik Software (linke Seite) finden Sie das ICON Software installieren/löschen. Bitte das Programm Software installieren/löschen mit einem Doppelklick aufrufen (siehe auch Installation der Datenbank). Über dieses Programm installieren Sie folgende Softwarepakete:</p> <div><table><tr><th>Paket</th><th>Größe</th><th>Beschreibung</th></tr><tr><td>postgresql-server</td><td>3,25 MB</td><td>Der Datenbank-Server zu PostgreSQL</td></tr><tr><td>sapdb-ibd</td><td>13,13 MB</td><td>SAP DB Kommunikationsserver und Be</td></tr><tr><td>sapdb-srv</td><td>60,60 MB</td><td>SAP DB Server</td></tr><tr><td>unixODBC</td><td>1,94 MB</td><td>ODBC-Treiber und Treibermanager</td></tr></table></div> <p>Abbildung 46 unixODBC-Softwarepakete</p> <ol style="list-style-type: none">1. unixODBC aus dem Paket Produktivität/Datenbanken/Server (siehe Bild oben) (nur bei ODBC!)2. postgresql-odbc aus dem Paket Produktivität/Datenbanken/Clients (nur bei ODBC!)3. unixODBC-gui-qt aus dem Paket Produktivität/Datenbanken/Tools (nur bei ODBC!)4. postgresql aus dem Paket Produktivität/Datenbanken/Tools (wenn das Programm psql gewünscht wird)	Paket	Größe	Beschreibung	postgresql-server	3,25 MB	Der Datenbank-Server zu PostgreSQL	sapdb-ibd	13,13 MB	SAP DB Kommunikationsserver und Be	sapdb-srv	60,60 MB	SAP DB Server	unixODBC	1,94 MB	ODBC-Treiber und Treibermanager
Paket	Größe	Beschreibung														
postgresql-server	3,25 MB	Der Datenbank-Server zu PostgreSQL														
sapdb-ibd	13,13 MB	SAP DB Kommunikationsserver und Be														
sapdb-srv	60,60 MB	SAP DB Server														
unixODBC	1,94 MB	ODBC-Treiber und Treibermanager														

Schritt	Inhalt / Aktion
3	Melden Sie sich nun an Ihrem Host ab und melden Sie sich an dem Host als Anwender root wieder an. Wenn Sie nur das Programm psql benötigen ist hier schon die Installation beendet.
4	Sie rufen danach das Programm ODBCConfig auf, in dem Sie über die Start -Taste, Büroprogramme , Datenbanken auf den Programmeintrag kommen.
5	<p>Rufen Sie in dem Programm die Karteikarte Drivers auf und drücken den Button Add.</p>  <p><i>Abbildung 47 unixODBC-Data Source Administrator</i></p>

Schritt	Inhalt / Aktion
6	<p>In dem nächsten Window arbeiten Sie folgende Punkte ab:</p>  <p style="text-align: center;"><i>Abbildung 48 unixODBC-Treiber</i></p> <ol style="list-style-type: none"> 1. Name vergeben: z.B. PGSQL 2. Description vergeben: z.B. für den POSTgre Server 3. Als Driver verwenden Sie die Datei /usr/lib/libodbcpsql.so 4. Als Setup-Datei nehmen Sie die Datei /usr/lib/libodbcpsqlS.so 5. Im Eintrag FileUsage lassen Sie die 1 stehen <p>ANMERKUNG: Die Dateien libodbcpsql.so und libodbcpsqlS.so können sich auch in einem anderen Verzeichnis befinden (siehe z.B. bei der Installation einer neuen Version aus dem Internet - Anhang C). In der SuSE 8.2 Version gab es ähnlich lautende Dateien, Sie sollten nur die oben gezeigten Dateien verwenden.</p>
7	<p>Danach rufen Sie die Karteikarte System DSN auf und drücken den Button Add</p>

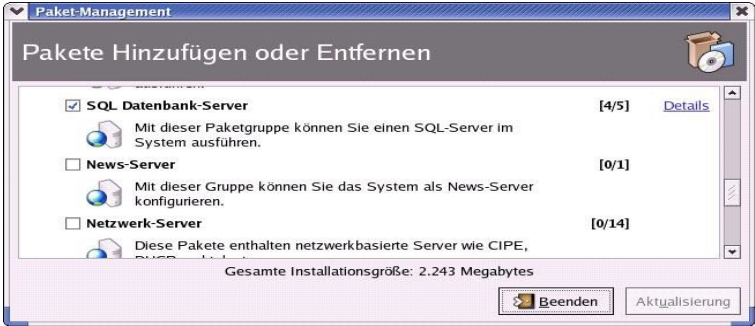
Schritt	Inhalt / Aktion
8	<p>Bearbeiten Sie im Window Data Source Properties folgende Punkte:</p>  <ol style="list-style-type: none"> 1. Namen vergeben 2. Description vergeben 3. Auswahl des Treibers (Driver) 4. Trace und TraceFile sind nicht unbedingt notwendig 5. Eingabe der Database 6. Eingabe des Servernamens 7. Eingabe eines UserNamens 8. Das Password für den User noch eingeben 9. Die Port-Nr. in der Zeile Port noch überprüfen <p><i>Abbildung 49 unixODBC-Datenquelle</i></p> <p>Danach alles wie gewohnt abspeichern und das Programm verlassen.</p>


Schritt	Inhalt / Aktion
9	<p>Nun kann man auch mit dem Programm DataManager (befindet sich direkt neben dem Programm ODBCconfig) auf die Datenbank zugreifen.</p>  <p><i>Abbildung 50 unixODBC-Treibermanager</i></p>

Anmerkung: Sollte die Verbindung von einem Client auf einen Host nicht möglich sein, so überprüfen Sie unter anderem die Einstellungen der Firewall auf dem Server (z.B. Sie haben die PersonalFirewall im Einsatz).

Red Hat/Fedora Distribution

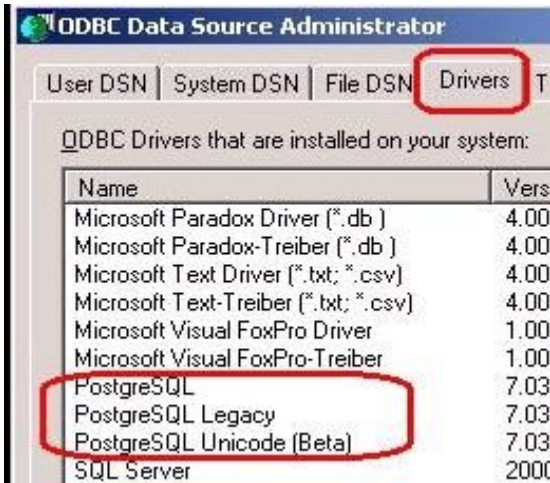
Bei der Linux-Distribution von Red Hat (ohne unixODBC) gehen Sie wie unten beschrieben vor (sie gleicht der Server-Installation und Sie erhalten auch die Server-Programme):


Schritt	Inhalt / Aktion
1	Drücken Sie die Start-Taste (mit dem roten Hut). Danach geht es weiter mit den Menüpunkten Systemeinstellungen und Hinzufügen / Entfernen von Applikationen
2	<p>Im Fenster Pakete Hinzufügen oder Entfernen finden Sie den Eintrag SQL Datenbank-Server. Wählen Sie diesen Punkt aus.</p> <div data-bbox="269 568 1024 895"></div> <p><i>Abbildung 51 unixODBC-Red Hat Softwarepakete</i></p>

Schritt	Inhalt / Aktion
3	<p>Wenn Sie im vorherigen Schritt auf den Eintrag Details drücken, sehen Sie im Detail, welche Software mit den Standardpaketen installiert wird.</p>  <p>Abbildung 52 unixODBC-Red Hat Softwarepakete</p>
4	<p>Danach schließen Sie die Auswahl ab und legen die geforderten CDs ein.</p>

Der ODBC-Treiber für Windows-Systeme

Sie finden die neuesten ODBC-Treiber auf der Homepage von PostgreSQL (www.postgresql.org). Über den Explorer rufen Sie die MSI-Datei auf (die Schritte sind hier etwas gepackt dargestellt) (**Anmerkung:** Auch bei dem Windows.NET Server 2003 gibt es keine Probleme).

Schritt	Inhalt																						
1	Installieren Sie hier die entsprechende MSI-Datei auf dem entsprechenden PC (psqlodbc.msi)																						
2	<p>Wechseln Sie in der Systemverwaltung in den ODBC Data Source Administrator. Unter der Karteikarte Drivers sollten Sie dann die PostgreSQL-Treiber finden.</p>  <table><thead><tr><th>Name</th><th>Vers</th></tr></thead><tbody><tr><td>Microsoft Paradox Driver (*.db)</td><td>4.00</td></tr><tr><td>Microsoft Paradox-Treiber (*.db)</td><td>4.00</td></tr><tr><td>Microsoft Text Driver (*.txt; *.csv)</td><td>4.00</td></tr><tr><td>Microsoft Text-Treiber (*.txt; *.csv)</td><td>4.00</td></tr><tr><td>Microsoft Visual FoxPro Driver</td><td>1.00</td></tr><tr><td>Microsoft Visual FoxPro-Treiber</td><td>1.00</td></tr><tr><td>PostgreSQL</td><td>7.03</td></tr><tr><td>PostgreSQL Legacy</td><td>7.03</td></tr><tr><td>PostgreSQL Unicode (Beta)</td><td>7.03</td></tr><tr><td>SQL Server</td><td>2000</td></tr></tbody></table> <p><i>Abbildung 53 Windows ODBC-Treiber</i></p>	Name	Vers	Microsoft Paradox Driver (*.db)	4.00	Microsoft Paradox-Treiber (*.db)	4.00	Microsoft Text Driver (*.txt; *.csv)	4.00	Microsoft Text-Treiber (*.txt; *.csv)	4.00	Microsoft Visual FoxPro Driver	1.00	Microsoft Visual FoxPro-Treiber	1.00	PostgreSQL	7.03	PostgreSQL Legacy	7.03	PostgreSQL Unicode (Beta)	7.03	SQL Server	2000
Name	Vers																						
Microsoft Paradox Driver (*.db)	4.00																						
Microsoft Paradox-Treiber (*.db)	4.00																						
Microsoft Text Driver (*.txt; *.csv)	4.00																						
Microsoft Text-Treiber (*.txt; *.csv)	4.00																						
Microsoft Visual FoxPro Driver	1.00																						
Microsoft Visual FoxPro-Treiber	1.00																						
PostgreSQL	7.03																						
PostgreSQL Legacy	7.03																						
PostgreSQL Unicode (Beta)	7.03																						
SQL Server	2000																						

Schritt	Inhalt
3	<p>Richten Sie danach entweder unter der User DSN oder System DSN den Zugriff auf die Datenbank ein. Sie geben hier die Datenbank (1), den Server (2) (IP-Adresse geht auch!), Benutzername (3) und Passwort (4) ein.</p>  <p><i>Abbildung 54 Windows ODBC-Einstellungen</i></p>
4	<p>Prüfen Sie z.B. über Access, ob Sie über die nun neu eingerichtete ODBC-Datenquelle auf die Tabellen des Host zugreifen können.</p>

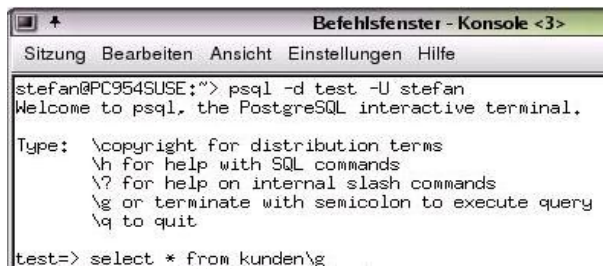
Das Programm psql

Allgemeines zum Programm psql

Mit diesem Programm können Sie verschiedene SQL-Befehle und Abfragen auf der Datenbank ausführen. Wenn Sie das Programm als Anwender **postgres** aufrufen, dann befinden Sie sich im Administrator-Modus (Sie erhalten das #-Zeichen als Prompt, sonst sehen Sie den Pfeil nach rechts), ansonsten befinden Sie sich im normalen User-Modus. Der vollständige Programmaufruf sieht so aus:

psql [Option] [dbname] [username]

Die Optionen sind im Text weiter unten beschrieben. Die Information **dbname** steht für die Datenbank und der Wert **username** beinhaltet den Benutzernamen. Weiterhin finden Sie auch die Beschreibung der Terminaloptionen.



```
Befehlsfenster - Konsole <3>
Sitzung Bearbeiten Ansicht Einstellungen Hilfe
stefan@PC954SUSE:~> psql -d test -U stefan
Welcome to psql, the PostgreSQL interactive terminal.

Type: \copyright for distribution terms
      \h for help with SQL commands
      \? for help on internal slash commands
      \g or terminate with semicolon to execute query
      \q to quit
stefan@PC954SUSE:~> select * from kunden\g
```

Abbildung 55 psql-Programmstart

WICHTIG: Mit der Zeichenkombination **\q** beenden Sie die Sitzung. Die Ausführung eines SQL-Kommandos veranlassen Sie durch die Buchstabenkombination **\g** (mit einem Semikolon den Befehl abschließen geht auch)! Wenn Sie Benutzer auf der Datenbank anlegen, dann sollten Sie als LINUX-Anwender **postgres** das Programm **psql** aufrufen.

Parameter	Inhalt
[options]	Die Terminaloptionen können Sie beim Start des Programms mit angeben (z.B. Sie führen eine SQL-Anweisung aus). Folgende Optionen gibt es:
-a	Gibt alle Eingaben aus dem Script aus
-c <query>	Führt eine einzige Abfrage (oder \-Befehl) aus und beendet psql
-d <dbname>	Bei einer bestimmten Datenbank anmelden. Fehlt der Datenbankname, dann wird der Anwendername als Datenbank verwendet.
-e	Gibt das SQL-Kommando als Überschrift aus (z.B. wenn SQL-Anweisungen aus einer Datei ausgeführt werden.
-f <filename>	Führt die Abfrage aus einer Datei aus und beendet das Programm.
-F <string>	Setzt den Datenfeldtrenner (default: " ") (-P fieldsep=)
-h <host>	Beinhaltet den Hostnamen einer Datenbank (wichtig , wenn Sie psql von einem anderen Host aufrufen)
-H	Wechselt in den HTML-Tabellen Ausgabemodus
-l	Zeigt die vorhandenen Datenbanken an und beendet anschließend das Programm
-o <filename>	Leitet die Ausgabe der Abfrage in eine Datei oder Pipe um
-p <port>	Enthält die Port-Nr. der Datenbank
-q	Läuft im Hintergrund, gibt keine Meldungen und Abfragen aus
-R <string>	Setzt den Datensatzseparator (default: Zeilenschaltung) (-P recordsep=)
-s	Der „kleine Debug-Modus“. Jede SQL-Anweisung sehen Sie und müssen die Ausführung bestätigen
-t	Zeigt nur die Spalten an (-P tuples_only)
-U <username>	Enthält den Benutzernamen beim Programmstart
-v name=val	Setzen einer psql-Variablen 'name' mit dem Wert 'val'
-V	Zeigt nur die aktuellen Versionsinformationen an
-x	Liest nicht die Konfigurationsdateien beim Start (~/.psqlrc)

Parameter	Inhalt
-d [dbname]	Der Name der Datenbank (TIPP: Die Datenbank template1 ist immer vorhanden)
-U [User]	Der Benutzername

Die Terminaloptionen des Programms psql

Sie können die hier gezeigten Befehle nur innerhalb des Programms psql ausführen!

Option	Inhalt
\a	Steuert den Ausgabemodus bei Abfragen, mit bzw. ohne Feldname
\c[onnect] [DBNAME]-[USER]	Auf einer neuen Datenbank anmelden
\C TITLE	Wird im Zusammenhang mit dem Select-Befehl verwendet. Gibt bei der Ausgabe zuerst den Titel und dann die Feldnamen aus
\c [DIRNAME]	Wechselt das aktuelle Verzeichnis
\copyright	Ausgabe der Copyright-Informationen
\d [TABELLE]	Ausgabe der Feldbeschreibungen einer Tabelle, wenn Sie die Tabelle weglassen, sehen Sie alle Tabellen
\da [Suchmuster möglich]	Ausgabe aller Aggregat-Funktionen
\dt [Suchmuster möglich]	Ausgabe alle Tabellen einer Datenbank
\di [Suchmuster möglich]	Ausgabe aller Indexe einer Datenbank
\ds [Suchmuster möglich]	Ausgabe aller Sequenzen einer Datenbank
\dv [Suchmuster möglich]	Ausgabe aller Views einer Datenbank
\dp [Suchmuster möglich]	Ausgabe aller Zugriffsrechte, Rechte sind: arwdRxt

Option	Inhalt
\dS [Suchmuster möglich]	Ausgabe aller Systemtabellen
\dl	Ausgabe aller „großen Objekte“
\dd [Suchmuster möglich]	Nach Kommentaren für Tabellen, Typen, Funktionen oder Operatoren sehen
\df [Suchmuster möglich]	Ausgabe aller Funktionen
\do [Name]	Ausgabe aller Operatoren
\dT [Suchmuster möglich]	Ausgabe aller Datentypen
\du [Suchmuster möglich]	Ausgabe aller Anwender
\e [DATEINAME]	Bearbeiten einer externen Datei im Editor vi
\echo [TEXT]	Schreiben eines Textes in die Standardausgabe
\f [STRING]	Setzen eines Feldtrenners
\g [DATEINAME]	Ausführen eines SQL-Kommandos aus einer Datei und Ausgabe in die Datei des SQL-Kommandos
\h [SQL-Befehl]	Hilfe über die Syntax eines SQL-Befehl anfordern, z.B. \h ALTER USER
\H	Wechselt in den HTML Ausgabemodus (standardmäßig aus)
\i [DATEINAME]	Ausführen eines Kommandos aus einer Datei, Ausgabe des Ergebnisses auf den Bildschirm
\l	Anzeigen aller Datenbanken
\o [DATEINAME]	Ausgabe aller Abfrageergebnisse in eine Datei oder in eine pipe
\p	Ansehen des Abfragepuffers (z.B. nachdem eine Select-Abfrage ausgeführt wurde)
\q	Das Programm psql verlassen
\r	Den Abfragepuffer löschen
\s [FILENAME]	Zeigt die bisher ausgeführten Befehle an. Eine Ausgabe in eine Datei ist möglich
\set [NAME] [VALUE]	Setzen einer internen Variablen

Option	Inhalt
\t	Nur die Zeilen einer SELECT-Anweisung ansehen
\unset [NAME]	Löschen einer internen Variablen
\w [DATEINAME]	Den aktuellen Abfragepuffer in eine Datei schreiben
\x	Wechseln des aktuellen Ausgabemodus. Man kann z.B. bei einer Select-Anweisung die Daten sequentiell oder tabellarisch ausgeben.
\z	Ausgabe der Zugriffsberechtigungen
\! [BEFEHL]	Ein Befehl oder eine Shell ausführen



Abbildung 56 psql-Liste verlassen

ANMERKUNG: Wenn sich die Hilfe über mehrere Zeilen erstreckt, dann können Sie über die Buchstabenkombination **:q** (wie beim vi) die Hilfe beenden.

Verwalten von Benutzern, Gruppen und Rechte

Um für die Datenbank Benutzer, Gruppen und Rechte zu verwalten, können Sie als LINUX-Anwender postgres das Programm `psql` aufrufen. Anschließend verwenden Sie zur Administration SQL-Befehle.

Benutzer verwalten

SQL-Befehle

Aktion	SQL-Befehl
Benutzer anlegen	<i>CREATE USER [NAME] PASSWORD 'geheim'</i> ANMERKUNG: das Wort geheim steht nur stellvertretend für das individuelle Passwort des Anwenders. Zwischen dem Wort PASSWORD und 'geheim' befindet sich ein Leerzeichen.
Vorhandenen Benutzer löschen	<i>DROP USER [NAME]</i>
Vorhandenen Benutzer ändern	<i>ALTER USER [NAME] [WITH] [Option, z.b. PASSWORD 'geheim', oder VALID UNTIL 'absolute Zeit im Format TT.MM.JJ']</i> ANMERKUNG: Es gibt noch mehrere Optionen.

ANMERKUNG: Mit `\du` (Programm `psql`) können Sie sich alle User anzeigen lassen

Auch von der Kommandozeile aus können sie Benutzer anlegen und löschen. Dazu haben Sie die Programme ***createuser*** und ***dropuser***. Beide Programme senden SQL-Befehle an die Datenbank. Sie setzen auch Superuser-Rechte voraus.

Programm createuser

Wenn noch keine Anwender angelegt sind, dann melden Sie sich als Unix-Anwender postgres an. Unter seiner Benutzerkennung legen Sie die weiteren Anwender an. Das Programm createuser ist wie folgt aufgebaut:

createuser [Optionen] [Benutzername]

Die verschiedenen Optionen des Programmes createuser entnehmen Sie der Tabelle (in den eckigen Klammern finden Sie die Kurzform der Option):

<i>Option</i>	<i>Inhalt</i>
--adduser [-a]	Der Benutzer kann neue Anwender anlegen
--no-adduser [-A] (großes A)	Der Benutzer hat nicht das Recht andere Anwender anzulegen
--createdb [-d]	Der Benutzer kann eine neue Datenbank anlegen
--no-createdb [-D]	Der Benutzer kann keine neue Datenbank anlegen
--pwprompt [-P]	Dem neuen Benutzer auch ein Passwort zuweisen
--encrypted [-E]	Das gespeicherte Passwort verschlüsseln
--unencrypted [-N]	Das gespeicherte Passwort nicht verschlüsseln
--sysid=SYSID [-i]	Dem neuen User auch eine SYSID in der Datenbank zuweisen
--echo [-e]	Die Meldungen bei der Aktion mit ausgeben (hier sehen Sie die SQL-Befehle)
--quiet [-q]	Die Meldungen bei der Aktion nicht ausgeben
--help	Die Hilfe für den Befehl

Wenn Sie keine Option angeben, dann fragt Sie das Programm nach den wichtigsten Optionen (siehe auch Bilder auf der nächsten Seite).



Abbildung 57 createuser ohne Optionen



Abbildung 58 createuser mit Optionen

Auch von Ihrer localem Host aus besteht die Möglichkeit auf anderen Hosts Anwender anzulegen. Dazu haben Sie diese Optionen (in den eckigen Klammern finden Sie die Kurzform der Option):

Option	Inhalt
--host=HOSTNAME [-h]	Der Name des Hosts
--port=PORT [-p]	Die Port-Nr. der Datenbank
--username=USERNAME [-U] (großes U)	Der Benutzername, mit dem Sie sich auf dem Datenbankserver anmelden
--password [-W] (großes W)	Das Passwort zu dem Benutzer, mit dem Sie sich auf dem Datenbankserver anmelden

Wenn Sie für mehrere Optionen die Kurzform wählen, dann nehmen Sie als Trenner das Leerzeichen (z.B. **createuser -h pc1300linux -D -A ralf**)

ANMERKUNG: Im Programm psql sehen Sie die angelegten Benutzer mit dem Befehl \du. Im unten gezeigten Beispiel wurde dem Benutzer janny die SYSID 555 zugewiesen und das Recht neue Benutzer anzulegen. Das Attribut „create database“ fehlt dagegen.



Abbildung 59 createuser - Benutzer abfragen

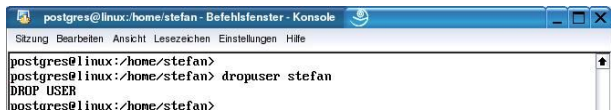
Programm dropuser

Nachdem Sie nun Benutzer angelegt haben, möchten Sie auch Datenbank-Anwender löschen. Dafür haben Sie das Gegenstück zum Programm createuser - dropuser -. Der Befehlsaufbau sieht so aus:

dropuser [OPTION]... [USERNAME]

Ihnen stehen folgende Optionen zur Verfügung (in den eckigen Klammern finden Sie die Kurzform der Option):

<i>Option</i>	<i>Inhalt</i>
--echo [-e]	Zeigt die SQL-Befehle an, die an die Datenbank gesendet werden.
--interactive [-i]	Abfrage, bevor gelöscht wird
--quiet [-q]	Gibt keine Meldungen aus
--host=HOSTNAME [-h]	Der Name des Hosts
--port=PORT [-p]	Die Port-Nr. der Datenbank
--username=USERNAME [-U] (großes U)	Der Benutzername, mit dem Sie sich auf dem Datenbankserver anmelden
--password [-W] (großes W)	Das Passwort zu dem Benutzer, mit dem Sie sich auf dem Datenbankserver anmelden

A screenshot of a terminal window titled "postgres@linux:/home/stefan - Befehlsfenster - Konsole". The window shows the following commands and output:

```
postgres@linux:/home/stefan>  
postgres@linux:/home/stefan> dropuser stefan  
DROP USER  
postgres@linux:/home/stefan>
```

Abbildung 60 dropuser - Beispiel

Gruppe verwalten

Verschiedene Benutzer kann man in einer Gruppe zusammenfassen. Wie auch bei den Anwendern verwalten Sie die Gruppe über SQL-Befehle.

Aktion	SQL-Befehl
Gruppe anlegen	CREATE GROUP [NAME]
Einen oder mehrere Anwender einer Gruppe zuordnen	ALTER GROUP [NAME] ADD USER USERNAME1, ... ANMERKUNG: Die 3 Punkte stehen für weitere Anwender. Verwenden Sie das Komma als Trennzeichen.
Einen oder mehrere Anwender aus einer Gruppe löschen	ALTER GROUP [NAME] DROP USER USERNAME1, ... ANMERKUNG: Die 3 Punkte stehen für weitere Anwender. Verwenden Sie das Komma als Trennzeichen.

Zugriffsrechte/Privilegien verwalten

Bei der Anlage einer neuen Datenbank wird ihr ein Owner (es ist der Anwender der die Datenbank erstellt hat) zugeordnet. Der Owner kann zurzeit nicht geändert werden. Erst durch die Erteilung von Zugriffsrechten können die anderen Anwender auf die Datenbank zugreifen. Mit dem SQL-Befehl GRANT verwalten Sie die Zugriffsrechte. Folgende Rechte gibt es:

Aktion	das dazugehörige Recht	Objekt
Lesen	Select	Tabelle
Hinzufügen	Insert	Tabelle
Schreiben	Update	Tabelle
Löschen	Delete	Tabelle
Anlegen	Create	Datenbank
Anlegen von temporären Tabellen	Temporary / Temp	Datenbank

Um einem Anwender ein Recht zu erteilen, ist der Befehl GRANT wie folgt aufgebaut:

```
GRANT [RECHT] ON  
[DATABASE Datenbankname | TABLE Tabellennamen]  
TO [BENUTZERNAME]
```

Wenn es geklappt hat, dann gibt das Programm zur Bestätigung nochmals das Recht aus. Um einer Gruppe ein Recht zu erteilen, ist der Befehl GRANT so aufgebaut:

```
GRANT [RECHT] ON  
[DATABASE Datenbankname | TABLE Tabellennamen]  
TO GROUP [GRUPPENNAME]
```

Geben Sie als Anwendernamen PUBLIC an, dann erteilen Sie jedem Anwender in der Datenbank das entsprechende Recht. Das Recht ALL beinhaltet übrigens alle Rechte.

Neben dem Zuerkennen von Rechten existiert der Befehl REVOKE um einem Anwender auch Rechte zu entziehen. Möchten Sie z.B. allen Anwendern alle Rechte auf einer Datenbank entziehen, dann geben Sie den Befehl REVOKE ALL ON [DATENBANK] FROM PUBLIC ein. Die Befehle DROP, GRANT, REVOKE usw. können im Bezug auf die Datenbank nur vom Owner ausgeführt werden und dem Owner auch nicht entzogen, bzw. anderen Benutzer zuerkannt werden. Der Owner einer Tabelle kann sich aber selber Rechte entziehen.

Anhang A - SQL-Befehle

Ändern	
Datensätze abändern	<p><code>Update [Tabellenname] set [Feld] = [Wert],... Pflicht where [Feld] [Bedingung] [Vorgabe] kann</code></p> <p>Wenn es mehrere Felder gibt, dann ist das Komma das Trennzeichen. Rechen-Operationen sind auch möglich z.B. <code>set Feld1 = Feld1 * 1,16</code>. Gültige Operationen sind <code>+</code> <code>-</code> <code>*</code> <code>/</code> <code>.</code></p>
Tabellenname ändern	<code>Alter table [Tabellenname] to [Tabellenname neu]</code>
Feld umbenennen	<code>Alter table [Tabellennamen] rename [Feld vorher] to [Feld nachher]</code>

Erstellen	
Tabelle anlegen	<p><code>Create table [Tabellenname] ([Feld] [Art])</code></p> <p>Wenn es mehrere Felder gibt, dann kommt direkt nach der Art des Feldes ein Komma. Folgende Feldtypen gibt es: <code>char(n)</code> => Zeichenfeld, n ist die Anzahl der Bytes <code>smallint</code> => + und - Ganzzahl <code>integer</code> => + und - Ganzzahl <code>decimal(n,m)</code> => Dezimalzahl (n = Zahl vor dem Komma, m = Zahl nach dem Komma) <code>money</code> => Betragsfeld</p>
Anlegen eines Indexfeldes	<p><code>Create unique distinct index [Indexname] on [Tabelle] (Feld,...)</code></p> <p>Der Index kann sich auch über mehrere Felder erstrecken <code>unique</code> => der Schlüssel wird nur einmal vergeben <code>distinct</code> => der Schlüssel darf mehrmals vorkommen</p>
Erstellen einer Datenbank	<p><code>Create database [Datenbank] [ENCODING='LATIN1']</code></p> <p>ANMERKUNG: Mit dem Shell-Befehl <code>createdb dbname</code> legen Sie auch eine Datenbank an. Ohne zusätzliche Option ist der aktuell angemeldete Benutzer der Owner. Übrigens auch das Encoding nicht vergessen.</p>

Ein- und hinzufügen	
Eingeben von Daten-sätzen	<p><code>Insert into [Tabellenname] ([Feld],...) values ([Wert],...)</code></p> <p>Anstelle der Felder und Werte kann man auch über eine select Anweisung verwenden.</p>
Einfügen von Feldern in eine Tabelle	<p><code>Alter table [Tabellenname] add column [Feld] [Feldtyp]</code></p>

Auswählen	
Von Daten	<p><code>Select [Feld],... Pflicht</code> <code>from [Tabellenname] Pflicht</code> <code>[join] kann</code> <code>where [Auswahlliste schränkt die Auswahl ein] kann</code> <code>group by [Gruppierungsliste fasst Feldinhalte zusammen] kann</code> <code>order by [Sortierliste sortiert die Ausgabe] kann</code> <code>into [Tabellenname die Ausgabe erfolgt in eine extra Tabelle]</code> <code>kann</code></p> <p>Anstelle der Felder kann auch ein Stern als Wildcard eingesetzt werden. Die Gruppierungs- und Sortierliste enthält die Feldnamen.</p> <p>ANMERKUNG: Möchten Sie nur eine bestimmte Anzahl von Datensätzen angezeigt haben, dann geben Sie nach der Tabelle LIMIT und die Anzahl der Datensätze ein. Z.B. bei SELECT * FROM KUNDEN LIMIT 10 erhalten Sie die ersten 10 Datensätze angezeigt. Mit der Option OFFSET <Anzahl Zeilen> überspringen Sie eine Anzahl von Zeilen..</p> <p>Sie können auch mit dem Select -Befehl plsql-Funktionen aufrufen, z. B. mit select now() das aktuelle Datum und die aktuelle Zeit.</p>

Löschen	
Löschen eines Indexfeldes	<code>Drop index [Indexname]</code>
Löschen einer Datenbank	<code>Drop database [Datenbank]</code> ANMERKUNG: Mit dem Befehl <code>dropdb dbname</code> als Shell-Befehl löschen Sie auch eine Datenbank
Löschen einer Tabelle	<code>Drop table [Tabellenname]</code>
Löschen eines oder mehrerer Datensätze	<code>Delete from [Tabellenname] Pflicht where [Bedingung] kann</code> Mit der Where-Bedingung schränkt man die zu löschenden Datensätze ein. ACHTUNG: Fehlt die Where-Bedingung, dann löscht man alle Datensätze.
Löschen eines Feldes in einer Tabelle	<code>Alter table [Tabellenname] drop column [Feld]</code>

Daten importieren und exportieren	
Import	<code>copy [binary] Tabellenname from 'Dateiname' delimiters '[das Zeichen]' [with null as 'Ersatzzeichen']</code> ANMERKUNG: Als Dateiname können Sie auch stdin angeben.
Export	<code>copy [binary] Tabellenname to 'Dateiname' delimiters '[das Zeichen]' [with null as 'Ersatzzeichen']</code> ANMERKUNG: Als Dateiname können Sie auch stdout angeben. Lassen Sie übrigens den Zusatz <code>WITH NULL AS</code> weg, dann gibt das Programm <code>\n</code> anstelle von null aus.

Autocommit	
Starten	<code>commit</code>
Beenden	<code>begin</code>

ANMERKUNG: Das Autocommit bewirkt, dass die Änderung sofort in die Datenbank geschrieben wird. Ohne diese Möglichkeit müssen Sie vor der Transaktion den Befehl `BEGIN` absetzen, dann Ihre Änderungen vornehmen und mit `COMMIT` dann festschreiben.

Agenda	
[Felder],...	Eingabe eines oder mehrerer Felder, bei mehreren Feldern muss man die Felder mit einem Komma trennen.
[Bedingung]	<p>Eine Bedingung setzt sich aus einem Feld, einer Bedingung und einem Vergleichswert zusammen. Gültige Bedingungen sind:</p> <ul style="list-style-type: none"> = (gleich) like (Suche nach Textmustern in einem Feld, % (bei ACCESS steht es für *) und ? sind als Wildcards zulässig) <> (ungleich) < (kleiner) <= (kleiner gleich) > (größer) >= (größer gleich) <p>Eine Bedingung kann auch mehrere Felder enthalten. Siehe Beispiele unten:</p> <p>PLZ="70565" => <i>ein Feld</i> KUNDENGRUPPE ="A" AND ORT ="ESSEN" => <i>zwei Felder</i></p>

Join	<p>Über join verknüpfen Sie Tabellen miteinander. Es gibt insgesamt 3 Arten von join:</p> <ul style="list-style-type: none">– CROSS JOIN: (z.B. Tabelle1 CROSS JOIN Tabelle2) Hier verknüpft die Abfrage jede Zeile aus der Tabelle1 mit der Tabelle2.– INNER JOIN: (z.B. Tabelle1 INNER JOIN Tabelle2) INNER ist die Grundeinstellung. Hier erhalten Sie alle Datensätze aus den Tabellen1 und Tabellen2, welche die Abfragebedingung erfüllen.– LEFT OUTER JOIN: z.B. Tabelle1 LEFT OUTER JOIN Tabelle2) Alle Datensätze aus der Tabelle1, die nicht die Abfragebedingung mit der Tabelle 2 erfüllen, werden ausgegeben.– RIGHT OUTER JOIN: z.B. Tabelle1 RIGHT OUTER JOIN Tabelle 2) Hier erhalten Sie alle Datensätze aus der Tabelle2, die nicht die Verknüpfungsbedingung mit der Tabelle1 erfüllen.– FULL OUTER JOIN: z.B. Tabelle1 FULL OUTER JOIN Tabelle2) Sie sehen hier alle Datensätze, die überhaupt nicht die Abfragebedingung erfüllen.
------	---

Anhang B – Anmerkung zum Datenbankentwurf

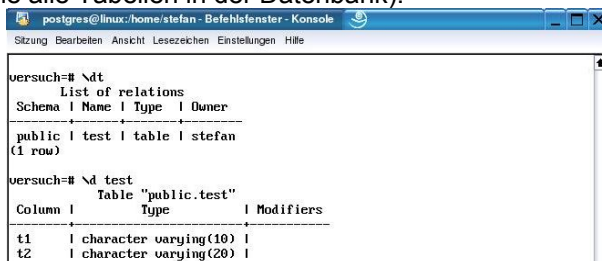
Datentypen

Von anderen Datenbanken sind Ihnen auch einige Datentypen bekannt. In der unten gezeigten Tabelle finden Sie die einzelnen Datentypen. Als Anwender können Sie übrigens auch eigene Datentypen anlegen.

Name	Alias	Beschreibung
bigint	int8	8-Byte-Ganzzahl mit Vorzeichen
bigserial	serial8	selbstzählende 8-Byte-Ganzzahl
bit		Bitkette mit fester Länge
bit varying(<i>n</i>)	varbit(<i>n</i>)	Bitkette mit variabler Länge
boolean	bool	Boole'scher (logischer) Wert (wahr/falsch)
box		Rechteck
bytea		binäre Daten, mit variabler Länge
character varying(<i>n</i>)	varchar(<i>n</i>)	Zeichenkette mit variabler Länge
character(<i>n</i>)	char(<i>n</i>)	Zeichenkette mit fester Länge
cidr		IP-Netzwerkadresse
circle		Kreis in der Ebene
date		Kalenderdatum (Jahr, Monat, Tag)
double precision	float8	Fließkommazahl mit doppelter Präzision
inet		IP-Hostadresse
integer	int, int4	4-Byte-Ganzzahl mit Vorzeichen
interval(<i>p</i>)		Zeitspanne
line		Gerade in der Ebene (nicht voll implementiert)
lseg		endliche Strecke in der Ebene
macaddr		MAC-Adresse
money		Geldbetrag

Name	Alias	Beschreibung
numeric [(p, s)]	decimal [(p, s)]	exakte Zahl mit wählbarer Präzision
path		offener oder geschlossener Pfad in der Ebene
point		geometrischer Punkt in der Ebene
polygon		Polygon
real	float4	Fließkommazahl mit einfacher Präzision
smallint	int2	2-Byte-Ganzzahl mit Vorzeichen
serial	serial4	selbstzählende 4-Byte-Ganzzahl (z.B. lfd. Nr. beim Einfügen in die Datenbank)
text		Zeichenkette mit variabler Länge
time [(p)] [without time zone]		Tageszeit
time [(p)] with time zone	timetz	Tageszeit mit Zeitzone
timestamp [(p)] without time zone	timestamp	Datum und Zeit
timestamp [(p)] [with time zone]	timestamptz	Datum und Zeit mit Zeitzone

TIPP: Wenn Sie sich den Aufbau einer Tabelle aus dem Programm `psql` ansehen möchten, dann geben Sie den Befehl `\d <Tabellenname>` ein (mit `\dt` sehen Sie alle Tabellen in der Datenbank).



```

postgres@linux:/home/stefan - Befehlsfenster - Konsole
Sitzung Bearbeiten Ansicht Lesezeichen Einstellungen Hilfe

versuch=# \dt
          List of relations
 Schema | Name | Type | Owner
-----+-----+-----+-----
 public | test | table | stefan
(1 row)

versuch=# \d test
          Table "public.test"
 Column |          Type          | Modifiers
-----+-----+-----
 t1     | character varying(10) | 
 t2     | character varying(20) | 

```

Abbildung 61 Tabellenaufbau anzeigen

Schemas

Nach der Neuanlage hat unsere Datenbank noch eine flache Struktur. Wir können z.B. alle Tabellen und Funktionen, die mit der Abteilung Verkauf zu tun haben, in einem Schema ablegen. Im Gegensatz zu der Datenbank sind die Schemas nicht strikt von einander getrennt. Wenn Sie kein Schema anlegen, dann greifen alle auf das Schema public zu.

Aktion	Inhalt
Schema anlegen	<code>CREATE SCHEMA name [AUTHORIZATION benutzername];</code> Mit AUTHORIZATION können Sie als DB-Administrator für jemand anderes anlegen, lassen Sie es weg, dann wird es mit dem aktuell angemeldeten Benutzernamen gleichgesetzt.
Zugriff auf eine Tabelle in einem Schema	<code>Schema.tabelle</code> oder <code>datenbank.schema.tabelle</code> .
Schema löschen	<code>DROP SCHEMA name</code> (geht nur wenn es leer ist, ansonsten: <code>DROP SCHEMA name CASCADE</code>)
Schema-Suchpfad	Mit der Anweisung SHOW search_path wird die aktuelle Einstellung ausgegeben. Voreingestellt sieht es so aus: \$user, public . Als erstes wird das Schema des Users durchsucht (soweit vorhanden) und dann das public. Mit der Anweisung SET search_path TO name,public; ändern wir die Einstellung vom Suchpfad.
Tabelle von einem Schema zu einem anderen Schema verschieben	Nur über den Weg Neuanlage der Tabelle im Zielschema und die Daten kopieren.

Anhang C - weitere Möglichkeiten zur Administration der Datenbank


Auf der Homepage www.postgres.de finden Sie unter der Rubrik Software eine Übersicht (incl. Link auf die Download-Bereiche) über verschiedene grafische Schnittstellen zur Datenbank. Von den Programmen habe ich nicht alle untersucht, sondern nur 2 ausgewählt. Sie finden dazu eine Screenshot und einen Verweis auf den Download-Bereich.

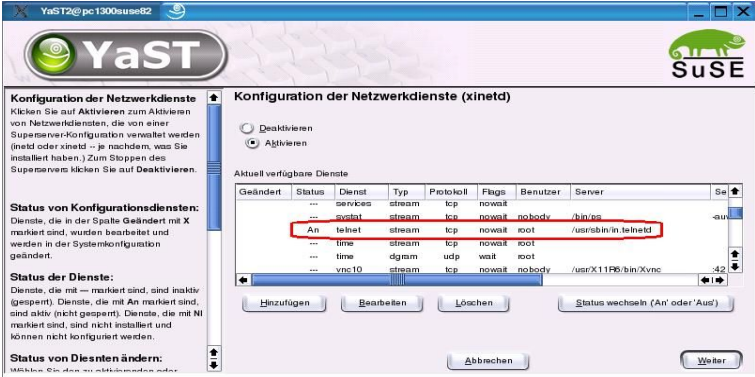
Administration über das Programm telnet

Das Programm finden Sie sowohl auf LINUX, als auch auf den MS-Windows Betriebssystemen. Dazu müssen Sie auf Ihrem LINUX-Host den Zugriff über **telnet** zulassen. Bei einer telnet-Sitzung können Sie nicht mit X-Windows arbeiten. Für die Einrichtung von **telnet** benötigen Sie wieder das Programm **YaST** (das Programm gibt es nur unter der SuSE Distribution). Die einzelnen Schritte finden Sie auf der nächsten Seite (**ANMERKUNG:** Fedora3 weist Sie daraufhin, dass Passwörter und Benutzernamen unverschlüsselt über das Netz gehen).

SuSE-LINUX und telnet

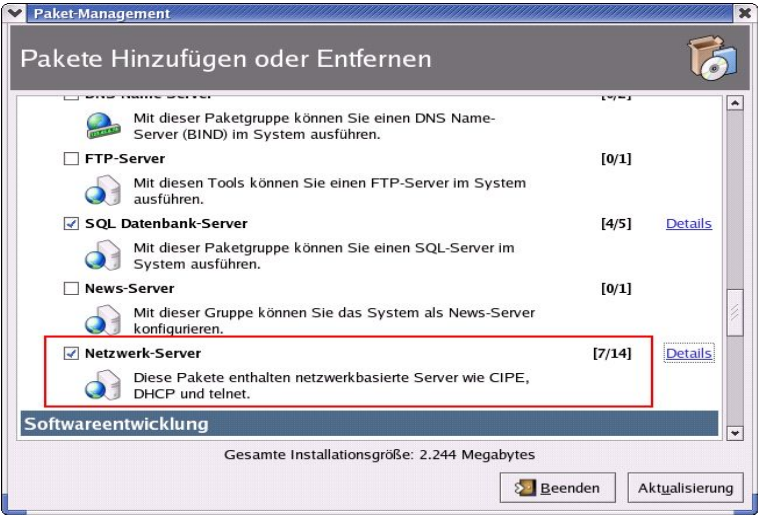
Unter SuSE-LINUX richten Sie es wie folgt ein:

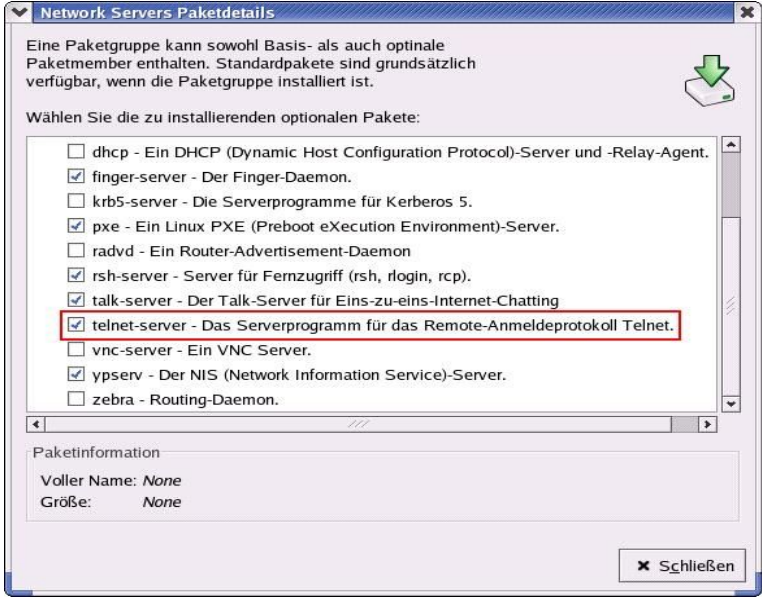
Schritt	Inhalt
1	 <p>Abbildung 62 SuSE und Telnet-Netzwerkdienste</p> <p>Wählen Sie hier Netzwerkdienste und Netzwerkdienste (inetd) aus (bei SuSE 8.2). Bei der Version 8.0 wählen Sie Netzwerk/Basis und Start oder Stop von Systemdiensten aus.</p>

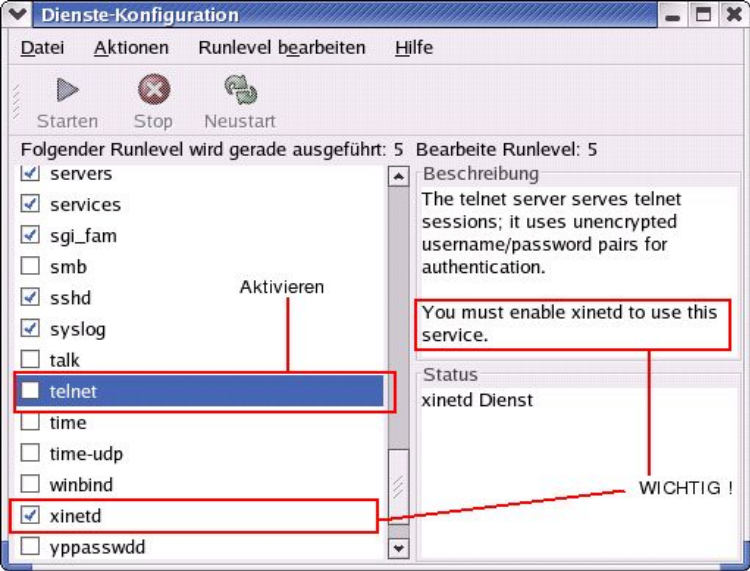
Schritt	Inhalt
2	<div></div> <p>Abbildung 63 SuSE und Telnet-Konfiguration</p> <p>Markieren Sie den Dienst telnet. Drücken Sie anschließend den Knopf Status wechseln ('An' oder 'Aus')</p> <p>WICHTIG: Achten Sie übrigens auch darauf, dass die Netzwerkdienste aktiviert sind!!</p>
3	<p>Prüfen Sie nun die Verbindungsaufnahme mit telnet. Wenn es nicht funktioniert, dann müssen Sie den inetd Dämon beenden und erneut starten (mit den Befehlen killall inetd und inetd als Superuser).</p>
4	<p>Weiter geht es nun mit dem Kapitel Mit telnet arbeiten.</p>

Red Hat/Fedora LINUX und telnet

Die Einrichtung des telnet-Servers gestaltet sich wie folgt:

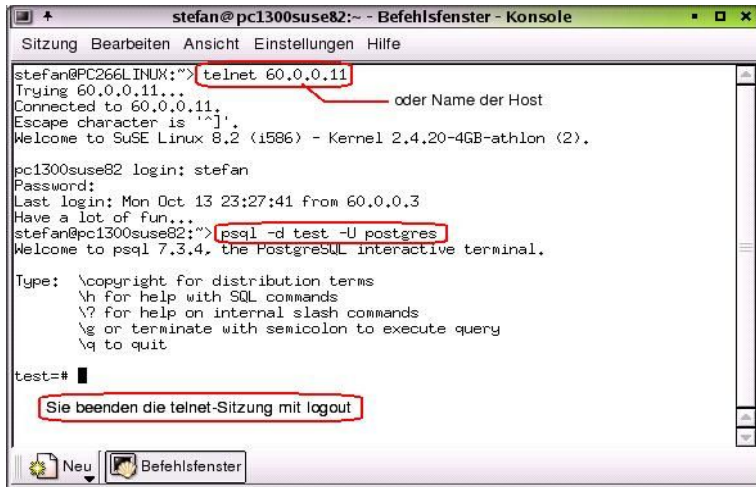
Schritt	Inhalt
1	Drücken Sie die Start-Taste (mit dem roten Hut). Danach geht es weiter mit den Menüpunkten Systemeinstellungen und Hinzufügen / Entfernen von Applikationen .
2	<p>Wählen Sie als nächsten Punkt den Eintrag Netzwerk Server aus (siehe unten). Überprüfen Sie auf jeden Fall, die Details zu diesen Paketen. Klicken Sie auf die Information Details von Netzwerk Server. Anschließend geht es beim Schritt 3 weiter.</p> <div data-bbox="262 529 1020 1046"></div> <p>Abbildung 64 Red Hat und Telnet-Pakete</p> <p>Bei Fedora3 finden Sie das Paket unter dem Eintrag Alte Netzwerk-Server.</p>

Schritt	Inhalt
3	<p>Der telnet-Server muss in diesem Fenster dann aktiviert sein.</p>  <p>Abbildung 65 Red Hat und Telnet-Serverprogramm</p> <p>Bei Fedora3 klicken Sie ebenfalls diesen Eintrag an.</p>
4	<p>Nun drücken Sie den Knopf Schließen und die Software wird auf Ihrem Host installiert. In der Fedora-Distribution sehen Sie noch einen Hinweis und drücken den Knopf Weiter.</p>

Schritt	Inhalt
5	 <p>Abbildung 66 Red Hat und Telnet - Dienste</p> <p>Im letzten Schritt aktivieren Sie noch beide Dienste (Start-Taste, Systemeinstellungen, Servereinstellungen). Bitte achten Sie auch auf den Dienst xinetd, da telnet ihn benötigt (siehe im Bild den Punkt WICHTIG).</p>
6	Weiter geht es nun mit dem Kapitel Mit Telnet arbeiten .

Mit Telnet arbeiten

Die Administration mit **telnet** gestaltet sich nun sehr einfach. Sie können hier mit allen textorientierten Programmen arbeiten.



```
stefan@pc1300suse82:~ - Befehlsfenster - Konsole
Sitzung Bearbeiten Ansicht Einstellungen Hilfe

stefan@PC266LINUX:~$ telnet 60.0.0.11
Trying 60.0.0.11...
Connected to 60.0.0.11.
Escape character is '^['.
Welcome to SuSE Linux 8.2 (i586) - Kernel 2.4.20-4GB-athlon (2).

pc1300suse82 login: stefan
Password:
Last login: Mon Oct 13 23:27:41 from 60.0.0.3
Have a lot of fun...
stefan@pc1300suse82:~$ psql -d test -U postgres
Welcome to psql 7.3.4, the PostgreSQL interactive terminal.

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

test=#
```

Sie beenden die telnet-Sitzung mit logout

Abbildung 67 Telnet anmelden

Nachdem Sie die Datenbank-Sitzung beendet haben, melden Sie sich mit dem Befehl **logout** von der LINUX-Host ab.

Administration über das Programm EMS PostgreSQL Manager

Das Programm finden Sie im Internet unter www.ems-hitech.com/pgmanager/download.phtml. Neben **pgAdmin** gibt es auch noch weitere Programme zur Administration der Datenbank. Es ist allerdings nicht kostenlos. Für 30 Tage können Sie es ohne Lizenzgebühren einsetzen. Im folgenden Bild sehen Sie einige Möglichkeiten, die das Programm bietet.

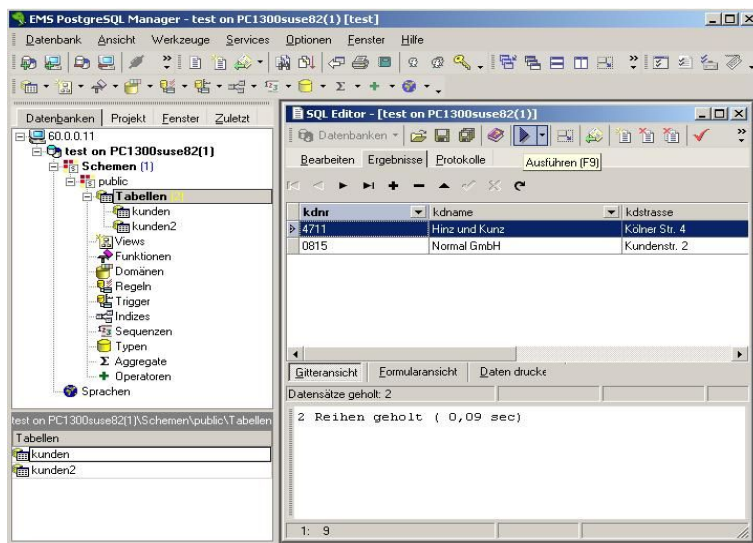


Abbildung 68 EMS PostgreSQL Manager

Administration über das Programm pgAdminIII

Das Programm finden Sie im Internet unter pgadmin.postgresql.org. Der pgAdminIII administriert nur Datenbanken ab der Version 7.3 (**ANMERKUNG:** Bei der Windows-Version 8.0 ist es schon dabei). Mit dem Programm pgAdminIII können Sie auch mehrere Datenbank-Server administrieren. Einen neuen Datenbank-Server fügen Sie sehr einfach hinzu. Markieren Sie einen Datenbank-Server oder den Eintrag Server. Drücken Sie die rechte Maustaste. Danach sehen Sie anschließend im Menue die Zeile **Server hinzufügen**.



Abbildung 69 pgAdmin III - Server hinzufügen

An einem Datenbank-Server melden Sie sich entweder durch einen Doppelklick auf den Eintrag an, oder auch über die rechte Maustaste (siehe unten).

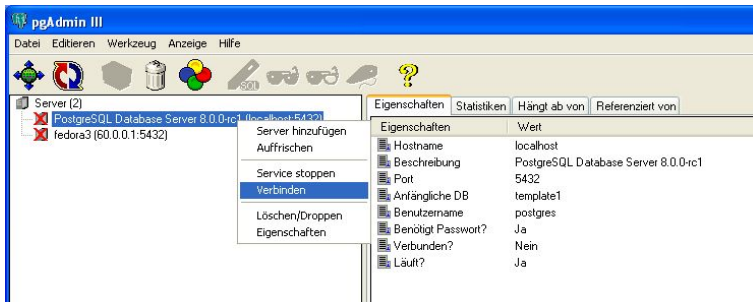


Abbildung 70 pgAdmin III - Anmelden

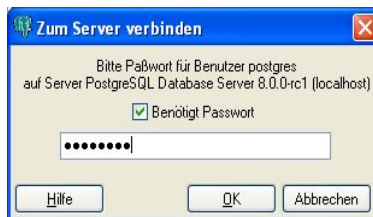


Abbildung 71 pgAdmin III-Eingabe Passwort

Die ersten Schritte in der Administration der PostgreSQL-Datenbank

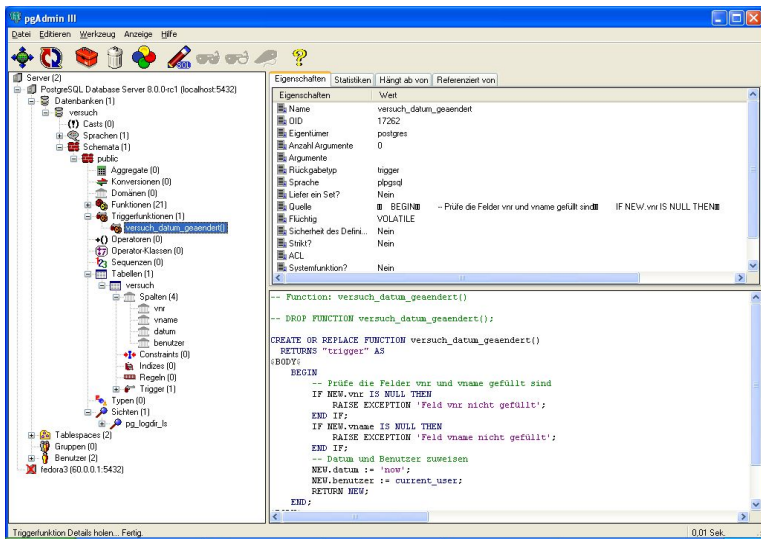


Abbildung 72 pgAdmin III Aufbau

Ähnlich wie beim Enterprise Manager vom MS SQL-Server können Sie sich nun durch den Baum hangeln. Generell gilt: Neue Einträge fügen Sie über die rechte Maustaste und den dann folgenden Menüeintrag hinzu.

Anhang D – die Zeichensätze

In dieser Anlage finden Sie die vom Server unterstützten Zeichensätze

<i>Name</i>	<i>Beschreibung</i>
SQL_ASCII	ASCII
EUC_JP	Japanischer EUC
EUC_CN	Chinesischer EUC
EUC_KR	Koreanischer EUC
JOHAB	Koreanischer EUC (Basis-Hangul)
EUC_TW	Taiwanischer EUC
UNICODE	Unicode (UTF-8)
MULE_INTERNAL	Mule Internal Code
LATIN1	ISO 8859-1/ECMA 94 (Lateinisches Alphabet Nr.1)
LATIN2	ISO 8859-2/ECMA 94 (Lateinisches Alphabet Nr.2)
LATIN3	ISO 8859-3/ECMA 94 (Lateinisches Alphabet Nr.3)
LATIN4	ISO 8859-4/ECMA 94 (Lateinisches Alphabet Nr.4)
LATIN5	ISO 8859-9/ECMA 128 (Lateinisches Alphabet Nr.5)
LATIN6	ISO 8859-10/ECMA 144 (Lateinisches Alphabet Nr.6)
LATIN7	ISO 8859-13 (Lateinisches Alphabet Nr.7)
LATIN8	ISO 8859-14 (Lateinisches Alphabet Nr.8)
LATIN9	ISO 8859-15 (Lateinisches Alphabet Nr.9)
LATIN10	ISO 8859-16/ASRO SR 14111 (Lateinisches Alphabet Nr.10)
ISO-8859-5	ISO 8859-5/ECMA 113 (Lateinisch/Kyrillisch)
ISO-8859-6	ISO 8859-6/ECMA 114 (Lateinisch/Arabisch)
ISO-8859-7	ISO 8859-7/ECMA 118 (Lateinisch/Griechisch)
ISO-8859-8	ISO 8859-8/ECMA 121 (Lateinisch/Hebräisch)
KOI8	KOI8-R(U)
WIN	Windows CP1251
ALT	Windows CP866
WIN1256	Windows CP1256 (Arabisch)
TCVN	TCVN-5712/Windows CP1258 (Vietnamesisch)
WIN874	Windows CP874 (Thai)

Anhang E - Liste der Dienstprogramme

In der Tabelle sind die Dienstprogramme aufgeführt. Einige von ihnen haben Sie ja schon kennengelernt.

<i>Dienstprogramm</i>	<i>Zweck</i>
clusterdb	Reorganisiert die Cluster einer Tabelle in einer Postgres Datenbank
createdb	Legt eine neue Datenbank an
createlang	Installiert eine prozedurale Sprache auf einer Datenbank
createuser	Legt einen Datenbank-Benutzer an
dropdb	Löscht eine Datenbank
droplang	Löscht eine prozedurale Sprache aus einer Datenbank
dropuser	Löscht einen Datenbank-Benutzer
initdb	Legt einen neuen Datenbank-Cluster an
pg_config	Gibt Informationen über die installierte Datenbankversion aus.
pg_controldata	Zeigt Informationen über den Server an
pg_ctl	Startet, beendet, oder führt einen Restart eines Datenbankservers durch
pg_dump	Für die Datensicherung einer einzelnen Datenbank. Die Ausgabe erfolgt in eine Datei
pg_dumpall	Für eine komplette Datensicherung. Das Programm schreibt <u>alle</u> Datenbanken in eine Datei
pg_resetxlog	Bereinigt den write-ahead log und den Inhalt der Datei pg_control
pg_restore	Stellt eine PostgreSQL Datenbank wieder her. Ausgangsbasis ist eine Datei, die von pg_dump erzeugt wurde
postgres	Betreibt die Datenbank im single-user mode
postmaster	Das Server-Programm erlaubt den Zugriff auf die Datenbank durch die verschiedenen Clients
psql	Ein Terminal-Programm für PostgreSQL
vacuumdb	Analysiert und bereinigt PostgreSQL Datenbanken

Quellen

1. Die Handbücher von Postgres, die im Lieferumfang enthalten sind
2. PostgreSQL Das offizielle Handbuch von Peter Eisentraut (von der PostgreSQL Global Development Group)
3. SuSE LINUX 8.0, die REFERENZ
4. Die Seite <http://www.postgres.de>
5. Linux von Michael Kofler
6. Zeitschrift iX – Dezember 2004, Artikel über PostgreSQL von Cornelia Boenigk und Ralf Burger

Abbildungsverzeichnis

Abbildung 1 SuSE YaST	11
Abbildung 2 SuSE Pakete auswählen	12
Abbildung 3 SuSE optimale Suche nach Paketen	12
Abbildung 4 SuSE Benutzerverwaltung	13
Abbildung 5 SuSE Benutzer anlegen	14
Abbildung 6 SuSE Benutzer und Gruppe	14
Abbildung 7 Red Hat Pakete Hinzufügen oder Entfernen Red Hat	15
Abbildung 8 Red Hat Paketdetails Red Hat	16
Abbildung 9 Red Hat Benutzerverwaltung	17
Abbildung 10 Red Hat Benutzereigenschaften Benutzerdaten	18
Abbildung 11 Red Hat Benutzereigenschaften Account-Info	18
Abbildung 12 Red Hat Benutzereigenschaft Passwort	19
Abbildung 13 Red Hat Benutzereigenschaft Gruppen	19
Abbildung 14 Datenbank einrichten	21
Abbildung 15 Datenbank starten	22
Abbildung 16 PGDATA	23
Abbildung 17 Skript zum Start der Datenbank	24
Abbildung 18 Dienst-Konfiguration	26
Abbildung 19 Datenbank beenden	29
Abbildung 20 postgresql.conf - Inhalt	33
Abbildung 21 Postmaster	35
Abbildung 22 Datenbanken anlegen - createdb	38
Abbildung 23 createdb - ein Beispiel	39
Abbildung 24 Datenbank Aktivitäten überwachen	53
Abbildung 25 Version 8 Windows, Installation, Sprache auswählen	54
Abbildung 26 Version 8 Windows, Installation	54
Abbildung 27 Version 8 Windows, Installation, Hinweise	55
Abbildung 28 Version 8 Windows, Optionen	55
Abbildung 29 Version 8 Windows, Dienste-Konfiguration	56
Abbildung 30 Version 8 Windows, User postgres	56
Abbildung 31 Version 8 Windows, Password	57
Abbildung 32 Version 8 Windows, Service Rechte	57
Abbildung 33 Version 8 Windows, database cluster	58
Abbildung 34 Version 8 Windows, remote Zugriff	58
Abbildung 35 Version 8 Windows, prozedurale Sprachen	59
Abbildung 36 Version 8 Windows, weitere Module	60
Abbildung 37 Version 8 Windows, Ende der Installation	61
Abbildung 38 Version 8 Windows, Dateien kopieren	61
Abbildung 39 Version 8 Windows, psql Hinweis	63
Abbildung 40 Version 8 Windows, CMD	63
Abbildung 41 Version 8 Windows, CMD Eigenschaften übernehmen	64

Abbildung 42	Version 8 Windows, Codepage	64
Abbildung 43	Version 8.0 Windows - Arbeiten mit psql	65
Abbildung 44	Version 8 Windows, Dienste	66
Abbildung 45	Version 8 Windows, Computerverwaltung	66
Abbildung 46	unixODBC-Softwarepakete	68
Abbildung 47	unixODBC-Data Source Administrator	69
Abbildung 48	unixODBC-Treiber	70
Abbildung 49	unixODBC-Datenquelle	71
Abbildung 50	unixODBC-Treibermanager	72
Abbildung 51	unixODBC-Red Hat Softwarepakete	73
Abbildung 52	unixODBC-Red Hat Softwarepakete	74
Abbildung 53	Windows ODBC-Treiber	75
Abbildung 54	Windows ODBC-Einstellungen	76
Abbildung 55	psql-Programmstart	77
Abbildung 56	psql-Liste verlassen	81
Abbildung 57	createuser ohne Optionen	84
Abbildung 58	createuser mit Optionen	84
Abbildung 59	createuser - Benutzer abfragen	84
Abbildung 60	dropuser - Beispiel	85
Abbildung 61	Tabellenaufbau anzeigen	94
Abbildung 62	SuSE und Telnet-Netzwerkdienste	97
Abbildung 63	SuSE und Telnet-Konfiguration	98
Abbildung 64	Red Hat und Telnet-Pakete	99
Abbildung 65	Red Hat und Telnet-Serverprogramm	100
Abbildung 66	Red Hat und Telnet – Dienste	101
Abbildung 67	Telnet anmelden	102
Abbildung 68	EMS PostgreSQL Manager	103
Abbildung 69	pgAdmin III - Server hinzufügen	104
Abbildung 70	pgAdmin III - Anmelden	104
Abbildung 71	pgAdmin III-Eingabe Passwort	104
Abbildung 72	pgAdmin III Aufbau	105