

Praktikumsaufgabe 1

Lernziele

Kennenlernen und Einrichten der Arbeitsumgebung (Postgres-Account, Umgebungsvariablen). Übung der Arbeit mit dem SQL-Interpreter und Schreiben von SQL-Scripts.

Vorbereitung

Folgende Informationen müssen Sie zum *Antestat* bereithalten:

- Informieren Sie sich über die elementaren Dateiverwaltungskommandos unter Linux (*ls, cd, rm, mkdir, mv, chmod*). ([1] Kap. 4)
- Wovon hängt es ab, ob eine ausführbare Datei vom System gefunden wird (Stichwort "PATH")? Wie fügt man ein Verzeichnis dieser Liste hinzu? ([1] Kap. 4)
- Wie zeigt man den Wert einer Umgebungsvariablen an? Wie setzt man eine Umgebungsvariable? Wo trägt man sie ein, damit sie bei jedem Login gesetzt ist? ([2] Kap. 8.6)
- Wie kommt man an die Online Hilfe zu Kommandos (Stichwort "Manpages")? ([1] Kap. 4)
- Wie startet man ein Programm im Hintergrund? ([1] Kap. 4)
- Wie legt man in SQL eine Tabelle an? Wie kann man Daten aus einer Tabelle in eine andere kopieren (Stichwort: *insert into ... select*)?
- Was machen die *psql* Metakommandos `\q`, `\d` und `\i`? ([3] Kap. "Reference, Client Applications")

Zur Arbeitsumgebung

Wenn Sie noch keinen Linux-Account haben, erhalten Sie im Praktikum einen Account XYZ und ein Kennwort. Damit können Sie sich an den Praktikumsrechnern anmelden. Sie können Ihr Kennwort mit dem Befehl *yppasswd* ändern. Sollten Sie Ihr Kennwort ändern, testen Sie bitte das neue Kennwort mit folgender Befehlsfolge ('\$' ist der Shell-Prompt):

```
$ su XYZ
Password:      # Achtung: Eingabe wird nicht angezeigt!
$ exit
```

Als Editor ist auf den Praktikumsrechnern *gedit* installiert, der bei Dateien mit der Endung *sql* automatisch die Syntax hervorhebt. Eine neue SQL-Datei legen Sie z.B. mit dem Befehl

```
$ gedit create.sql &
```

Aufgaben

1.1 Online-Hilfe. Stellen Sie sicher, dass Sie auf die PostgreSQL Online-Dokumentation zugreifen können.

1. Starten Sie das Eingabefenster und zeigen Sie sich die Hilfe von *psql* an mit den Befehlen (\$ steht für den Shell-Prompt und # ist ein Kommentarzeichen):

```
$ psql --help # gibt Benutzungshinweise von psql aus
$ man psql    # Manpage (beenden mit 'q', suchen mit '/')
```

2. Legen Sie im Webbrowser *firefox* ein Bookmark an auf die lokale PostgreSQL Dokumentation unter */usr/share/doc/postgresql-doc-8.4/html/index.html*

1.2 Datenbank-Account. Neben dem Unix-Account erhalten Sie einen Datenbank-Account. Testen Sie, ob Sie sich damit anmelden können mittels

```
$ psql -h host -U user database
```

Für *database* ist hierbei der Username zu verwenden. Für *host* ist der Name des Datenbankservers *glisrv-4* einzusetzen. Anschließend nehmen Sie bitte folgende Einstellungen vor:

1. Ändern Sie Ihr Kennwort in der Datenbank mit dem SQL-Befehl (“psql>” ist der *psql*-Prompt):

```
psql> ALTER USER user PASSWORD 'kennwort';
```

2. Wenn Sie die Werte für *host*, *database* und *user* nicht immer eingeben wollen, können Sie sie als Umgebungsvariablen *PGHOST*, *PGDATABASE*, *PGUSER* setzen. Wenn Sie die folgenden Einstellung in Ihrer Datei *\$HOME/.profile* vornehmen, werden die Variablen automatisch beim Einloggen gesetzt (Wichtig: Wenn Sie Änderungen an dieser Datei vorgenommen haben, testen Sie diese bitte erst im offenen Terminal mit dem Befehl “Punkt Leerzeichen Punkt profile”, d.h. “. *.profile*”):

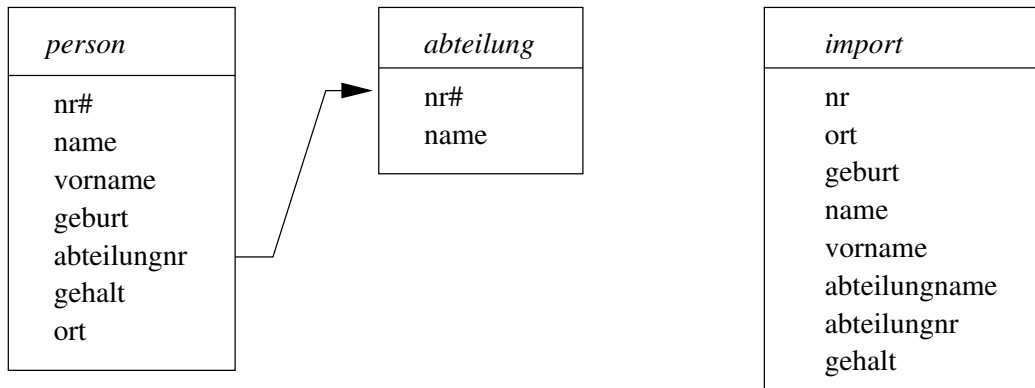
```
PGHOST=glisrv-4
PGUSER=dbsXY      # anpassen
PGDATABASE=$PGUSER
export PGHOST PGUSER PGDATABASE
```

1.3 psql. Diese SQL-Shell dient zum Senden von SQL-Befehlen an die Datenbank. Ferner können *psql*-interne Befehle (“Metakommandos”, beginnen alle mit einem Backslash) eingegeben werden. Die Dokumentation zu SQL finden Sie in der Postgres HTML-Doku. Dort gibt es auch einen Abschnitt zu *psql*, der der Manpage entspricht. Alle Metakommandos können auch in *psql* mit \? aufgelistet werden.

Führen Sie folgende Aufgaben durch:

1. Anlegen einer Tabelle mit nur einer Spalte direkt in *psql*
2. Anzeigen der Struktur dieser Tabelle (welches Metakommando?)
3. Erstellen eines SQL-Scripts *insert.sql* (Editorbefehl: *gedit*) mit Befehlen zum Befüllen der Tabelle und Ausführen dieses Scripts in *psql* (welches Metakommando?)
4. Entfernen der Tabelle

1.4 SQL. Erstellen Sie ein SQL-Skript zum Anlegen des folgenden Datenmodells:



Die Tabelle *import* wird nur vorübergehend für den Datenimport benötigt und sollte nur aus *varchar* Feldern bestehen.

Für die weiteren Aufgaben benötigen Sie die Datei *dbsnam-utf8.csv*, die Sie in Ihr Home-Verzeichnis mit folgendem Befehl kopieren können:

```
cd # wechselt ins Home-Verzeichnis
cp ../D999999/pub/dbsnam-utf8.csv .
```

Importieren Sie die Datei *dbsnam-utf8.csv* in die Tabelle *import* mit dem Befehl (Achtung: den Befehl *nicht* aus dieser PDF-Datei kopieren!)

```
\copy import from 'dbsnam-utf8.csv' delimiter ';' ;
```

Listen Sie auf wieviele Daten in *import* gelandet sind und testen Sie ob Umlaute richtig angezeigt werden (ggf. das Encoding vor dem Import auf *utf8* umstellen mit dem Metakommando *\encoding*).

Schreiben Sie ein SQL-Skript zum Kopieren der Daten aus der Tabelle *import* in die Zieltabellen (Hinweis: verwenden Sie für die Abteilungstabelle *select distinct*). Führen Sie anschließend folgende Abfragen *auf den Zieltabellen* durch:

- Aus welchem Ort kommen wieviele Mitarbeiter? (*group by*)
- Wie heißt die Abteilung, in der Norbert Herrling arbeitet? (*join*)
- Was ist das Durchschnittsgehalt je Abteilung? (*avg()* und *group by*)

Anmerkung

Sie dürfen auch Ihren eigenen Rechner im Praktikum verwenden. Beachten Sie aber bitte, dass Sie sich in diesem Fall selber um eine korrekte Installation und Einrichtung der Software kümmern müssen. In dieser Anmerkung gebe möchte ich nur einen wichtigen Hinweis geben für ein typisches Problem beim Zugriff auf die installierte Software.

Wenn Sie PostgreSQL auf Ihrem eigenen Rechner installiert haben und dabei eine Installation aus den Quellen gewählt haben, ist zu beachten dass die komplette Software standardmäßig in das Verzeichnis `/usr/local/pgsql` installiert wird, das sich nicht im Suchpfad für Programme, Bibliotheken oder Manpages befindet.

Durch folgende Einträge in der Datei `$HOME/.profile` können sie die entsprechenden Pfade bekannt machen:

```
PGROOT=/usr/local/pgsql          # zum Abkürzen
PATH=$PATH:$PGROOT/bin          # Suchpfad Executables
MANPATH=$MANPATH:$PGROOT/man    # Suchpfad Manpages
LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$PGROOT/lib # Suchpfad Libs
export PATH MANPATH LD_LIBRARY_PATH # weiterleiten an Subshells
```

Unter MacOS X ist die Variable `LD_LIBRARY_PATH` zu ersetzen durch `DYLD_LIBRARY_PATH`, so dass die letzten beiden Zeilen statt dessen lauten

```
DYLD_LIBRARY_PATH=$DYLD_LIBRARY_PATH:$PGROOT/lib # Suchpfad Libs
export PATH MANPATH DYLD_LIBRARY_PATH # weiterleiten an Subshells
```

Wenn Sie die Datenbank nur lokal auf Ihrem Rechner zum Testen und nicht produktiv nutzen, können Sie auch in der Postgres-Konfigurationsdatei `pg_hba.conf` die Authentifizierungsmethode "trust" für alle Benutzer wählen: dann kann sich jeder ohne Kennwort anmelden. In dem Fall sollten Sie natürlich nicht den Netzzugriff auf die Datenbank freischalten (das ist bei einer Installation aus den Quellen standardmäßig ausgestellt, kann aber ggf. bei Binärpaketen von Linux-Distributionen standardmäßig aktiv sein).

Referenzen

- [1] Welsh, Kaufmann: *Linux - Wegweiser zur Installation & Anwendung*. O'Reilly (1998)
Semesterapparat (TWR Wels)
- [2] Kofler: *Linux - Installation, Konfiguration, Anwendung*. Addison-Wesley, 7. Auflage (2006)
Semesterapparat (TWR Kofl)
- [3] The PostgreSQL Global Development Group:
PostgreSQL 8.3.9 Dokumentation. <http://www.postgresql.org/docs/> (2008)