

# Elefant im Blick

Der Datenbankzugriff per Kommandozeile ist nicht jedermanns Sache. Für mehr Durchblick sorgt PG-Access, eine praktische Oberfläche für PostgreSQL (Maskottchen ist ein Elefant). Sie unterstützt den Anwender bei typischen Aufgabe wie Tabellenerzeugung, Ex- und Import von Daten, Backup und Reporting. Carsten Zerbst



Der erste Schritt bei der Arbeit mit PG-Access ist das Verbinden des Tools mit einer oder mehreren Datenbanken.

Das gelingt entweder bereits beim Aufrufen über Kommandozeilenoptionen (siehe **Tabelle 1**) oder im laufenden Betrieb über das Menü »Database | Open«. Ist in PostgreSQL noch keine Datenbank angelegt, kann man sich zur Systemdatenbank »template1« verbinden.

PG-Access verwendet – wie andere Clients auch – eine TCP/IP-Verbindung zur Kommunikation mit PostgreSQL, deshalb muss der Server-Admin das Datenbanksystem mit der Option »-i« starten und den Zugang in der Datei »pg\_hba.conf« freigeben. Auf Wunsch speichert PG-Access die Verbindungsdaten auch, damit der Anwender sie nicht immer wieder neu eingippen muss.

Die Benutzerführung ist bei PG-Access einheitlich gelöst, **Abbildung 1** zeigt die deutschsprachige Oberfläche. Wer diese statt der voreingestellten englischen Varianten verwenden will, konfiguriert entsprechend das GUI über »Database | Preferences | Preferred Language«. PG-Access gibt daraufhin die Meldung »Die neuen Schrifteinstellungen werden erst beim nächsten Neustart wirksam« aus. An den Schriften hat sich zwar

nichts geändert, der Hinweis gilt aber auch für die Spracheinstellung. Der gleiche Konfigurationsdialog legt außerdem fest, ob PG-Access die Systemtabellen mit anzeigen soll.

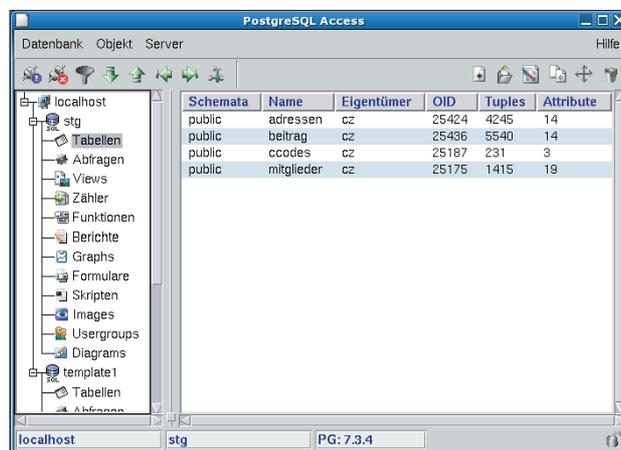
## Aufbau des Hauptfensters

Für jeden Funktionsbereich enthält die Baumdarstellung am linken Fensterrand einen eigenen Knoten (**Abbildung 1**): Ausgehend vom DB-Server über die dort vorhandenen Datenbanken navigiert man zu den Tabellen, Abfragen, Views und weiteren Bereichen. Die rechte Seite des Hauptfensters zeigt die jeweiligen Objekte, zum Beispiel die Tabellen oder die Benutzer.

Im »Objekt«-Menü (und im Kontextmenü) sind die Aktionen zu finden, die sich auf das ausgewählte Objekt auswirken: »Neu«, »Öffnen«, »Ändern«, »Umbenennen« und »Löschen«. In den meisten Fenstern findet sich eine Hilfefunktion. Die Hilfetexte sind größtenteils lesenswert, es gibt sie aber nur auf Englisch.

**PostgreSQL** zählt seit Jahren zu den besten freien Datenbanken. Die wenigsten Anwender möchten heute jedoch nur mit »psql« auf der Kommandozeile arbeiten, sie erwarten eine Anwendung mit grafischer Oberfläche. Constantin Teodorescu startete deshalb die Entwicklung von PG-Access [1]. Mit diesem Programm lassen sich die typischen Datenbankaufgaben wie Entwurf, Import und Export von Daten, Abfragen und vieles mehr in einer einfach zu bedienenden GUI-Umgebung lösen.

Inzwischen arbeiten mehrere Programmierer an der Weiterentwicklung der aktuellen Version 0.98.8. PG-Access ist in den meisten Distributionen enthalten. Wenn nicht, lässt es sich mit wenigen Handgriffen als fertig gepacktes Starkit (siehe **Kasten „Starkits“**) starten.



**Abbildung 1:** Das Hauptfenster von PG-Access stellt links eine Treeview dar, rechts die jeweiligen Objekte (hier die Tabellen in der Datenbank »stg«).

Tabellen sind das wesentliche Element von Datenbanken; PG-Access gibt einen schnellen Überblick über ihren Aufbau. Der Menüpunkt »Objekt | Entwurf« startet einen Dialog, der alle wesentlichen Informationen über die vorhandenen Tabellen gibt (Abbildung 2). Hier kann man – entsprechende Rechte vorausgesetzt – auch Spalten hinzufügen und Benutzerrechte ändern. Für weiter gehende Änderungen wie Spalten löschen und Referenzen setzen ist PG-Access allein nicht ausreichend, dazu sind normale

SQL-Kommandos nötig. Neue Tabellen lassen sich dagegen leicht mit der grafischen Oberfläche entwickeln, das Dialogfenster unterstützt alle Funktionalitäten bis auf Referenzen.

## Mit Diagrammen den Überblick behalten

Um einen schnellen Überblick oder ein Bild für die Dokumentation zu bekommen, bieten sich Diagramme an: Bereich »Diagrams« im Baum auswählen, dann

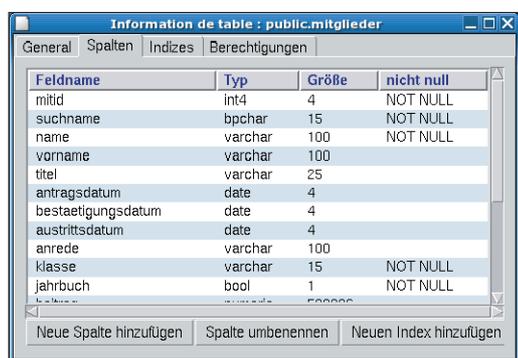


Abbildung 2: Die Entwurfsansicht für Tabellen gibt einen Überblick über ihre Eigenschaften, hier zeigt das Fenster die Spalten der Tabelle »public.mitglieder«.

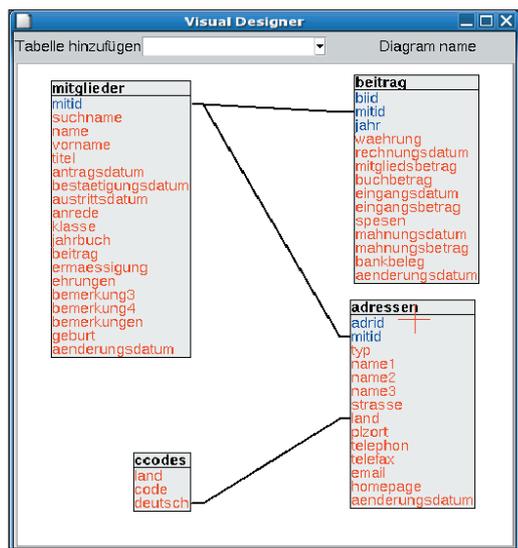


Abbildung 3: Im Visual Designer lassen sich einfache Diagramme zeichnen. Die Grafiken dienen nur zur Dokumentation des eigenen Datenbankentwurfs.

mit »Objekt | Neu« ein neues Diagramm anlegen. Dieser Editor (Abbildung 3) dient dazu, schnell UML-ähnliche Grafiken zu zeichnen. Referenzbeziehungen können mit der Maus angelegt werden: einfach Spaltennamen von einer Tabelle auf Spalten einer anderen Tabelle ziehen.

Der Punkt »Objekt | Öffnen« im Bereich »Tabellen« zeigt ihren Inhalt an, in der Defaultkonfiguration aber nur die ersten 200 Datensätze. Die Liste kann der Anwender durch Filter und Sortierung verfeinern. Die Ergebnisse lassen sich auch bequem exportieren, im Export-Wizard stehen CSV (Comma-separated Values), HTML-Tabellen und benutzerdefinierte Formate zur Verfügung.

Viele Probleme können mit diesem Dialog schon erledigt werden, ohne dass der Anwender Ahnung von SQL haben muss. Das gilt auch fürs Einfügen von Daten, die Tabellenfelder sind wie in einer Tabellenkalkulation editierbar. Größere Datenmengen fügt man besser über den Import-Wizard aus CSV oder HTML-Tabellen ein.

Je nach Problemstellung kommt bald der Tag, an dem eine richtige Datenbankabfrage abzusetzen ist. Man kann sie in PG-Access entweder per Hand eingeben oder grafisch entwickeln. Für die direkte Eingabe steht ein einfacher Editor zur Verfügung (siehe Abbildung 4), die Ergebnisse stellt PG-Access in einem eigenen Fenster dar. Das sieht zwar so aus wie die normale Tabellenansicht, leider lässt sich aber hier die Sortierung nicht ändern, die Filterfunktion ist ebenfalls nicht einsetzbar. ▶

### Starkits

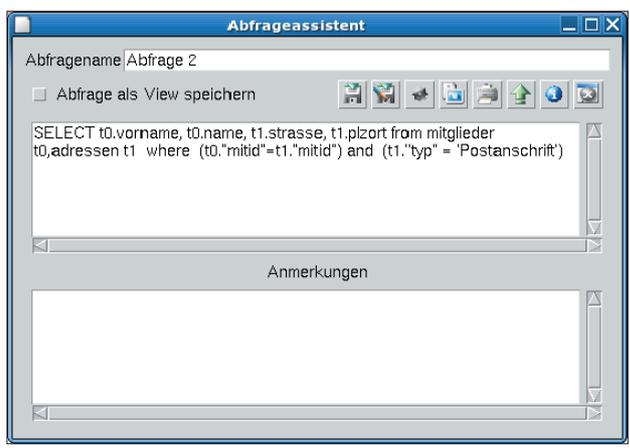
Egal ob Perl, Python oder Tcl - viele Anwendungen auf Skriptbasis kämpfen mit dem Problem, dass die von ihnen verwendeten Erweiterungen nicht bei allen Anwendern installiert sind. In Mode kommen verschnürte Pakete wie die Tcl-Starkits. Sie enthalten die komplette Anwendung inklusive der benötigten Erweiterungen. Um Starkits zu starten ist lediglich ein spezieller Interpreter (Tclkit) nötig, er steht unter [2] zur Verfügung. Diese Datei muss nur noch entpackt und ausführbar gemacht werden, danach startet Tclkit beliebige Starkits:

```
wget http://www.equi4.com/pub/tk/8.4.5/2
tclkit-linux-x86.gz
gunzip tclkit-linux-x86.gz
chmod 755 tclkit-linux-x86
wget http://www.bscharz.com/projects/2
pgaccess/pgaccess.kit
./tclkit-linux-x86 pgaccess.kit
```

Viele Anwendungen sind im Starkit-Distributionsarchiv [3] zu finden, eine ausführliche Beschreibung der dahinter stehenden Technik ist in [4] zu finden.

### Tabelle 1: Wichtige Optionen

Aufrufoption	Bedeutung
-host Adresse	Servername
-pgport Nummer	Serverport, per Default 5432
-username Name	Benutzername
-password Geheim	Passwort
-dbname Name	Datenbankname
-pgintcl	Eingebaute PostgreSQL-Erweiterung benutzen



**Abbildung 4:** Der Abfrageassistent nimmt ein SQL-Kommando entgegen und zeigt das Ergebnis in einem neuen Fenster. PG-Access kann den Suchstring direkt in der Datenbank ablegen, sodass mehrere Benutzer davon profitieren.

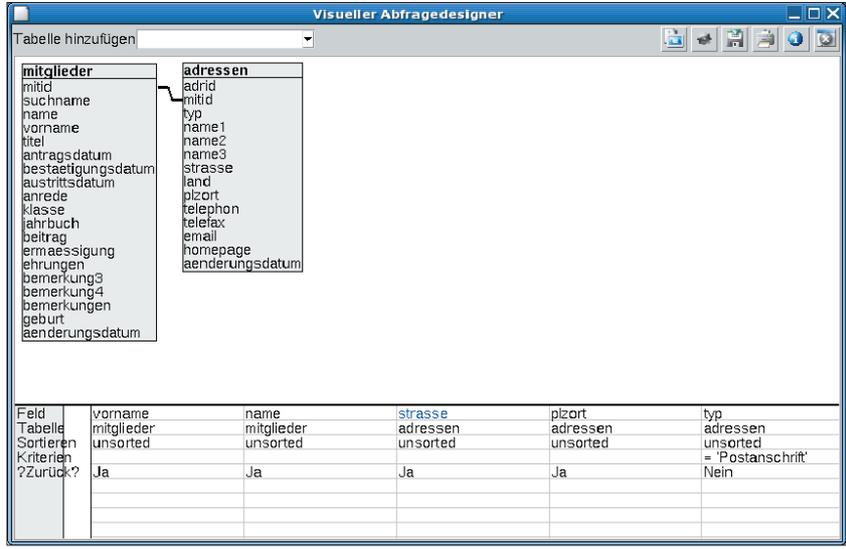
Fehler in den Abfragen behandelt das GUI robust und zeigt sie durch eine verständliche Fehlermeldung an. Für den Fall, dass mehr Information erforderlich ist, enthält die Onlinehilfe auch ein komplettes PostgreSQL-Handbuch. Häufig benötigte Abfragen kann der Anwender – wie viele andere PG-Access-Eingaben auch – für die spätere Wiederholung direkt in der Datenbank speichern. Sollen auch andere Benutzer die gespeicherten Abfragen in PG-Access verwenden, müssen die Rechte der »pga\_\*«-Tabellen ausreichend offen gesetzt sein. Gerade für Anfänger ist es leichter, eine Abfrage mit dem visuellen Abfragedesigner zusammenzustellen (Abbildung 5). Er wird aus dem normalen Abfragedialog gestartet (das vierte Icon) und ist

in weiten Bereichen durch Drag & Drop zu bedienen. Nachdem er per »Tabelle hinzufügen« alle notwendigen Tabellen geladen hat, kann der User die gewünschten Felder in den unteren Fensterbereich ziehen und dort Einschränkungen angeben. Sind bei einer Abfrage mehrere Tabellen zu verknüpfen, genügt es, die entsprechende Spalte aus der einen Tabelle auf die korrespondierende Spalte in einer anderen Tabelle zu ziehen. Nach dem Verlassen des Abfragedesigners steht die Abfragen leider nur noch im normalen Abfragedialog zur Verfügung, der visuelle Designer lädt sie nicht wieder.

### Tcl integriert

Abfragen dürfen bei PG-Access sogar Tcl-Skripte enthalten, die das Programm vor der Abfrage ausführt. Das folgende Skript sucht nach Personen, deren Name ist beim Ausführen in einem Dialogfenster einzugeben:

```
select * from mitglieder where ?
name=' [parameter "Name"] '
```



**Abbildung 5:** Mit dem visuellen Abfragedesigner lassen sich Queries einfacher entwickeln als mit SQL-Direktiven. In der oberen Hälfte sind die Tabellen und ihre Verknüpfungen zu sehen, unten die gewünschten Felder.

Mit dieser Funktion lassen sich leicht Abfragen für Benutzer vorbereiten, die nicht in den SQL-Kommandos editieren können oder sollen. Wer weiter gehende Ansprüche stellt, könnte sich innerhalb von PG-Access auch ganze Formulare erstellen, allerdings ist dieser Bereich kaum dokumentiert. Deshalb ist es zurzeit wahrscheinlich einfacher, komplett neue Applikationen zu schreiben, statt auf PG-Access aufzusetzen. Der Ex- und Import von CSV-Dateien kann und soll kein Backup ersetzen. Über den Menüpunkt »Server | Dump database« kann der Admin daher einen richtigen Dump erzeugen. Diese Dumps enthalten bis auf wenige Ausnahmen normales SQL, die Daten könnten somit auch auf andere Datenbanken umziehen. Unter »Server | PgMonitor« findet sich auch gleich ein Werkzeug, das den Status von PostgreSQL überwacht und zu lange dauernde Abfragen unterbricht, im schlimmsten Fall beendet es PostgreSQL sogar.

### Bedingt benutzerfreundlich

Für die üblichen Datenbankaufgaben zeigt sich PG-Access benutzerfreundlich, zum Beispiel durch die einfachen Abfragen und das bequeme Ändern von Tabellen. Sind die Anforderungen an die Logik etwas höher, kann eine Abfrage mit eingebettetem Tcl auch für Endanwender ohne SQL-Kenntnisse eine schnelle Lösung sein. Problematisch sind speziellere Bereiche wie beispielsweise Formulare oder Reports. Es ist fraglich, ob man als Programmierer nicht schneller ein eigenes Skript entwickelt – zumindest solange die Dokumentation diese Bereiche nicht besser erläutert. Gerade für aufwändig formatierte Reports bietet sich daher eher die Kopplung mit Open Office oder Star Office an. (fjl)

#### Infos

- [1] PG-Access: [\[http://www.pgaccess.org\]](http://www.pgaccess.org)
- [2] Tclkit: [\[http://www.equi4.com/tkdownload.html\]](http://www.equi4.com/tkdownload.html)
- [3] Starkit-Archiv: [\[http://mini.net/sdarchive/\]](http://mini.net/sdarchive/)
- [4] Carsten Zerbst, „Sterngucker: Tcl-Module zu einer ausführbaren Datei verschnüren“: Linux-Magazin 1/04, S. 100