

## Vorbereitungen für Aufgabenblätter 3, 4 und 5: PostgreSQL Installation

Für die Aufgabenblätter 3-5 benötigt ihr eine lokale Installation von PostgreSQL. Wir haben im folgenden einige Anleitungen zur Installation zusammengefasst. Sollte es bei euch zu Problemen bei der Installation kommen, versucht euch zunächst selbst zu behelfen (im Netz gibt es viele Anleitungen/Tutorials). Bei schwer lösbaren Problemen könnt ihr uns natürlich gerne kontaktieren. Für die Bearbeitung der Aufgaben braucht ihr lediglich die PSQL-Shell (Kommandozeile). Grafische Oberflächen sind nicht nötig (ihr dürft sie aber natürlich verwenden).

a) Hinweise zu Windows Setup und häufigen Problemen:

- Laden Sie den Installer z.B. von dieser Webseite runter:

<https://www.postgresql.org/download/windows/>

Während der Installation gibt es die Option, Stack Builder mit zu installieren. Das ist für diese Übung **nicht** notwendig.

Für die Übung wird nur die SQL Shell "psql" benötigt. Nach Start der Shell muss Enter gedrückt werden, bis man die Passwordeingabe erreicht. Danach sollte es folgendermaßen oder ähnlich aussehen wie in Abbildung 1

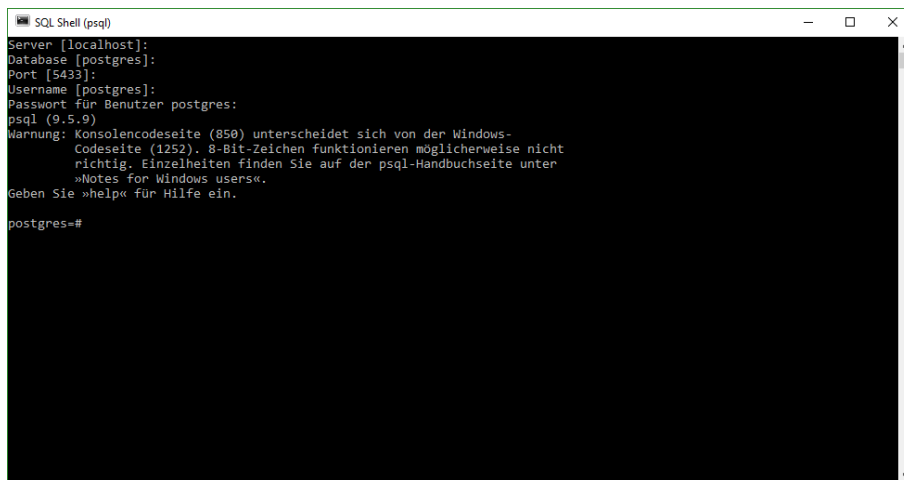


Abbildung 1: Windows SQL Shell nach erfolgreicher Installation und Passwordeingabe.

Zugriffsprobleme können im Allgemeinen durch die Ausführung als Administrator gelöst werden. Da der Backslash für Kommandos reserviert ist, müssen Backslashes in Pfaden durch Slashes ersetzt werden.

b) Hinweise zu Linux Setup (Ubuntu 16.04 LTS):

- Eine Shell öffnen und `sudo apt-get install postgresql-contrib-9.5` ausführen
- Nun `sudo -u postgres psql` um PostgreSQL unter dem User postgres zu starten
- Nun sind sie mit der psql Shell verbunden. Sobald Sie eine Datenbank angelegt haben können Sie sich mit dieser direkt aus dem Terminal mit `psql <dbname>` verbinden.

c) Hinweise zu Mac Setup:

- Für die Installation wird empfohlen, den beliebten Packagemanager Homebrew zu nutzen, Ablauf:
  - Terminal öffnen
  - `brew -v` um zu überprüfen, ob Homebrew bereits installiert ist, ggf. installieren siehe <https://brew.sh/>

- `brew install postgresql@9.6` um PostgreSQL 9.6 zu installieren
- am Ende des ausgegebenen Installationslogs wird empfohlen, PostgreSQL zum Path hinzuzufügen; für die Bash-Shell sieht das dann beispielsweise so aus: `echo export PATH="/usr/local/opt/postgresql@9.6/bin:$PATH" >> ~/.bashrc`
- Terminalsession beenden und eine neue starten, damit der neu hinzugefügte Path übernommen wird
- PostgreSQL-Installation ist damit abgeschlossen, ab jetzt kann man PostgreSQL folgendermaßen im Hintergrund starten und auch wieder stoppen
  - \* `brew services start postgresql@9.6`
  - \* `brew services stop postgresql@9.6` (beim Stoppen werden die angelegten Datenbanken *nicht* gelöscht)
- `psql postgres` um sich mit PostgreSQL zu verbinden
- `\q` um die Verbindung zu beenden

**Import der Daten: Siehe nächste Seite**

## Import der Daten

- Zip Datei Lehrveranstaltungen\FG Informationssysteme\VL DBS I\Übung 2019\data.zip herunterladen und entpacken
- PostgreSQL Kommandozeile öffnen
- Mittels `CREATE DATABASE imdb;` und `CREATE DATABASE sales;` zwei verschiedene Datenbanken erzeugen (mit `\c <dbname>` könnt ihr zu der entsprechenden Datenbank wechseln)
- Mittels `\i <pathToFile>` könnt ihr ein SQL-Skript ausführen. Die heruntergeladenen Skripte importiert Daten in die Datenbank mit der ihr gerade verbunden seid. Das Kommando `\d` zeigt anschließend fälschlicherweise noch keine Tabellen an. Dies wird behoben indem ihr euch mit einer beliebigen anderen Datenbank verbindet und anschließend wieder zurückwechselt. Führt die beiden Import-Skripte `imdb;import.sql` und `sales;import.sql` in den entsprechenden Datenbanken aus. Es kann sein, dass dies bei sales ein paar Minuten dauert.
- Nun könnt ihr testen ob alles funktioniert hat, indem ihr überprüft ob die in Abbildung 2 und 3 gezeigten Queries die gleichen Resultate liefern. Lediglich der Owner wird bei euch anders sein.

```
dbs1_imdb=# \d
List of relations
Schema | Name | Type | Owner
-----+-----+-----+-----
public | actor | table | leon
public | actress | table | leon
public | genre | table | leon
public | laptop | table | leon
public | movie | table | leon
public | moviexml | table | leon
public | producer | table | leon
(7 rows)

dbs1_imdb=# select count(*) from actress;
count
-----
3479
(1 row)

dbs1_imdb=# select * from movie where title = '4 aventures de Reinette et Mirabelle';
mid | title | year
-----+-----+-----
4 | 4 aventures de Reinette et Mirabelle (1987) | 1987
(1 row)
```

Abbildung 2: Test Queries with Output (IMDB)

```
sales=# select * from orders limit 10;
o_orderkey | o_custkey | o_orderstatus | o_totalprice | o_orderdate | o_orderpriority | o_clerk | o_shippriority | o_comment
-----+-----+-----+-----+-----+-----+-----+-----+-----
1 | 36991 | O | 173665.47 | 1996-01-02 | 5-LOW | Clerk#000000951 | 0 | instructions sleep furiously among
2 | 78902 | O | 46929.18 | 1996-11-01 | 1-HIGHEST | Clerk#000000850 | 0 | focus, pending accounts at the pending, silent asymptot
3 | 123314 | P | 193846.25 | 1993-10-14 | 5-LOW | Clerk#000000955 | 0 | sly final accounts boost, carefully regular ideas cajole carefully, depos
4 | 328777 | O | 23151.78 | 1995-10-11 | 5-LOW | Clerk#000000124 | 0 | site, slyly regular warehouse cajole, regular, regular theodolites acro
5 | 44485 | P | 144659.20 | 1994-07-30 | 5-LOW | Clerk#000000925 | 0 | quickly, bold deposits sleep slyly, packages use slyly
6 | 55024 | P | 58709.59 | 1992-02-21 | 4-NOT SPECIFIED | Clerk#000000950 | 0 | 20%r, special, final requests are against the furiously specta
7 | 39136 | O | 252094.18 | 1996-01-10 | 2-HIGH | Clerk#000000470 | 0 | ly special requests
32 | 130857 | O | 208668.75 | 1995-07-16 | 2-HIGH | Clerk#000000610 | 0 | use blithely bold, regular requests, quickly unusual dep
33 | 60958 | P | 152543.98 | 1993-10-27 | 3-MEDIUM | Clerk#000000400 | 0 | arduously, furiously final request
34 | 61801 | O | 58949.67 | 1998-07-21 | 3-MEDIUM | Clerk#000000223 | 0 | ly final packages, fluffily final deposits wake blithely ideas, spe
(10 rows)

sales=# select count(*) from orders;
count
-----
100000
(1 row)
```

Abbildung 3: Test Queries with Output (Sales)