

Übung 1: Einführung in PostgreSQL und pgAdmin

Ziele:

- Kennenlernen von PostgreSQL und pgAdmin als Administrationswerkzeug
- Verbinden zum PostgreSQL-Datenbank-Server
- Importieren einer Datenbank
- Erste SQL-Abfragen

Zeit:

- Ca. 50 Min.

Kennenlernen pgAdmin und PostgreSQL-Server:

- Starten Sie pgAdmin (bzw. pgAdmin4)
- Erstellen Sie eine Datenbank-Verbindung (Name ist z.B. „PostgreSQL 16 local“)
- Beantworten Sie folgende Fragen (evtl. mit Hilfe der Abbildung 1 unten):
 - Was ist im Navigations-Baum von pgAdmin (linke Spalte) alles dargestellt?
 - Welche globalen Objekte gibt es und welche Objekte gibt es per Datenbank?
 - Wo werden in der PostgreSQL-Architektur die Tabellen verwaltet?
 - Was ist das “public schema” und was ist ein „user defined schema”?
 - Probieren Sie die verschiedenen Kontext-Menüs (Rechts-Klick) auf Objekten im Navigations-Baum aus.
 - Das pgAdmin GUI kann man konfigurieren: blenden Sie z.B. alle FTS-DB-Objekte aus.
 - Starten Sie das separate SQL-Fenster mit dem man auf der Datenbank SQL-Befehle ausführen kann. Wie heisst dieses SQL-Fenster in pgAdmin?

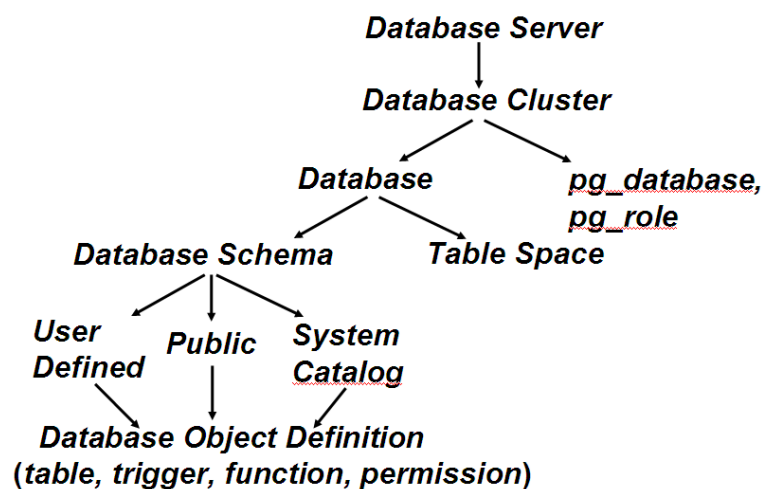


Abbildung: Architektur von PostgreSQL.

Aufgabe 1

Erstellen Sie eine neue Datenbank mit dem Namen „**postgis_kurs_uster**“.

Beachten Sie, dass sie alle Objektnamen in PostgreSQL klein schreiben sollten, ansonsten müssen Sie um jedes Objekt doppelte Hochkomma (") dazufügen. Sonderzeichen, mit Ausnahme des underscores "_" sollten ebenfalls vermieden werden. Verwenden Sie „UTF-8“ als default-Encoding. Verbinden Sie danach auf die neu erstellte Datenbank.

Importieren Sie von den Übungsdaten das **Backup-File „postgis_kurs_uster.backup“** (Rechts-Klick auf DB „postgis_kurs_uster“ → Restore). Der Import dauert etwas.

Hinweis: Der Import fügt die PostgreSQL-Extension PostGIS hinzu zur leeren Datenbank. Das kann man auch im pgAdmin tun mit „Create > Extension“. Oder man macht es mit SQL z.B. im (SQL) Query-Tool: `CREATE EXTENSION postgis;` .

Wenn die Extension nicht existiert (und nicht mit dem Import installiert wird), kommt beim Import/Restore ein Fehler sinngemäss z.B. „ERROR: function addgeometrycolumn(unknown...)...“.

Kontrollieren Sie ob im Schema „public“ die Tabelle „spatial_ref_sys“ und die View „geometry_columns“ etwas enthalten, z.B. mit „View/Edit Data (All Rows)“.

Überprüfen Sie, ob die Datenbank 17 Schematas (inkl. „public“-Schema) enthält.

Überprüfen Sie ob im Schema „av_user“ viele „views“ (Sichten) enthalten sind (49) und ob der View „av_user.gebaeude“ tatsächlich Daten enthält (Kontextmenü über Rechtsklick auf „av_user.gebaeude“ View im Navigationsbaum „Daten anzeigen → Alle Zeilen zeigen / View Data → View All Rows“ oder via SQL „select count(*) from av_user.gebaeude;“ (sollte ca. >8000 Gebäude enthalten).

Aufgabe 2

- Verbinden Sie sich mit der Datenbank „postgis_kurs_uster“ (wenn nicht schon getan).
- Testen Sie die installierte Version von PostgreSQL und PostGIS und die verfügbaren Datenbanken: Geben Sie die folgenden Befehle im SQL-Fenster ein (kann auch kleingeschrieben sein; "--" ist ein SQL-Kommentar):
 - Abfragen der PostgreSQL-Version: `SELECT version();`
 - Abfragen der PostGIS-Version: `SELECT postgis_full_version();`
 - Auflistung aller verfügbaren Datenbanken:
 - Mit pgAdmin im Panel links
 - Mit SQL: `SELECT datname, oid FROM pg_database ORDER BY oid;`
 - Auflistung aller installierten Extensions: `SELECT * FROM pg_extension;`
 - Sehen Sie sich die View-Definition von „av_user.gebaeude“ an und analysieren Sie wie der View aufgebaut ist. Welche Tabellen sind beteiligt und wie sind sie verknüpft?

Aufgabe 3

Finden Sie heraus in welchen Raumplanungszonen mit mehr als 50 Einwohnern (view „raumplanung.nutzungszonen“) der Frauenanteil grösser als 55% und der Ausländeranteil > 25% ist.

Tipp: Verschiedene WHERE-Bedingungen verknüpfen.

Aufgabe 4

Errechnen Sie den Prozentanteil der Bevölkerung in Uster die in Zonen mit erhöhter Lärmempfindlichkeit leben (Stufen III und IV)? Betrachten Sie dazu den View „raumplanung.nutzungszonen“. Relevant sind die Spalten „laermempfindlichkeit“ und „bev_gesamt“. Geben Sie das Resultat in %, gerundet auf eine Kommastelle aus.

Tipps:

- Subselects verwenden.
- Integer-Typ muss in numeric Typ umgewandelt werden. Es gibt dazu in PostgreSQL den „casting“-Operator „::“ (nicht SQL-Standard) sowie die standardisierte SQL-Funktion CAST.

Aufgabe 5

Selektieren Sie vom View „baustellen.vw_baustellen_projekte_linienelemente“ diejenigen Baustellen bei denen Strasse und Gas betroffen sind und die Baustelle zumindest teilweise zwischen Mai und September 2014 ausgeführt wird.

Tipps:

- Spalten „betroffen_strasse“, „betroffen_gas“, „start_baustelle“ und „ende_baustelle“. „to_date“ verwenden für Datumsvergleiche.
- Bedingungen mit AND und OR sowie Klammern verknüpfen.
- Für den Filter nach dem Zeitraum gibt es das SQL-Schlüsselwort OVERLAPS.

Lösungen

Aufgabe 3:

```
SELECT * FROM raumplanung.nutzungszonen
  WHERE bev_gesamt > 50 AND bev_anteil_ch < 75 AND bev_anteil_weiblich > 55;
```

Ergebnis: 2 Zonen (W3/70G 1, W4/70 II)

Aufgabe 4:

```
SELECT
round((SELECT SUM(bev_gesamt)::numeric FROM raumplanung.nutzungszonen WHERE
laermempfindlichkeit IN ('ES_III','ES_IV'))
/
(SELECT SUM(bev_gesamt)::numeric FROM raumplanung.nutzungszonen)
* 100,1) AS anteil_bevoelkerung_erhoehter_laerm;
```

Ergebnis: 33.6

Aufgabe 4:

```
SELECT *
FROM baustellen.vw_baustellen_projekte_linienelemente
  WHERE betroffen_strasse = 't' AND betroffen_gas = 't'
  AND ((start_baustelle, ende_baustelle) OVERLAPS ('2014-05-01', '2014-09-30'))
ORDER BY start_baustelle ASC, objekt_projektbezeichnung ASC;
```

Alternativ (ohne OVERLAPS Operator):

```
SELECT *
FROM baustellen.vw_baustellen_projekte_linienelemente
  WHERE betroffen_strasse = 't' AND betroffen_gas = 't'
  AND (
    (start_baustelle < '2014-05-01' AND ende_baustelle > '2014-09-30') OR
    (start_baustelle < '2014-05-01' AND ende_baustelle > '2014-05-01') OR
    (start_baustelle < '2014-09-30' AND ende_baustelle > '2014-09-30') OR
    (start_baustelle > '2014-05-01' AND ende_baustelle < '2014-09-30')
  )
ORDER BY start_baustelle ASC, objekt_projektbezeichnung ASC;
```

Ergebnis:

```
Burgstrasse
Greifenseestrasse - Verlängerung
Pfäffikerstrasse
Sonnenbergstrasse 2.Etappe
Tumigerstrasse
Weidstrasse
```