

Übung 4: Geometriedaten laden und darstellen

Ziele:

- Laden von Dateien in den Formaten Shapefile, CSV (GeoPackage, ...)
- Laden aus CSV-Datei, Update über WKT-Geometrieformat
- Darstellen von PostGIS-Daten in QGIS
- Filtern von Daten in QGIS

Zeit:

- Ca. 30 Min.

Aufgabe 1 Laden von Shapefiles:

Laden Sie mit Hilfe des Werkzeugs „PostGIS Shapefile Import/Export Manager“ (früher „shp2pgsql-gui“) die Shapefiles und damit die Tabellen „fluesse“, „gemeinden“ und „seen“ in das Schema „schweiz“ ihrer Datenbank „postgis_kurs_uster“. Setzen Sie dazu die richtigen Verbindungsparameter und Optionen.

Bemerkung: Shapefiles sind veraltet. Siehe <http://switchfromshapefile.org/> . GeoPackage sind der aktuelle Standard. In PostGIS wird leider immer noch nur dieses Werkzeug mitgeliefert – jedoch im Wissen, dass es beispielsweise GDAL und QGIS gibt.

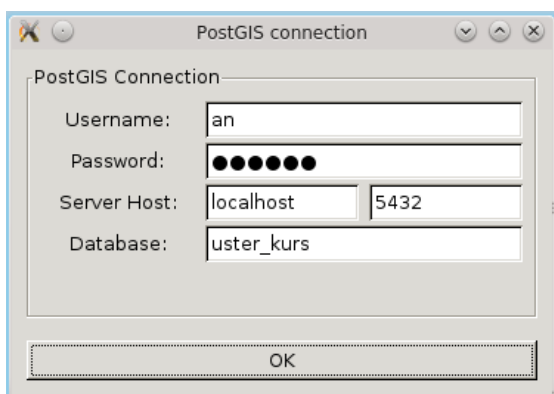


Abbildung 1: Verbindungsparameter von shp2pgsql-gui

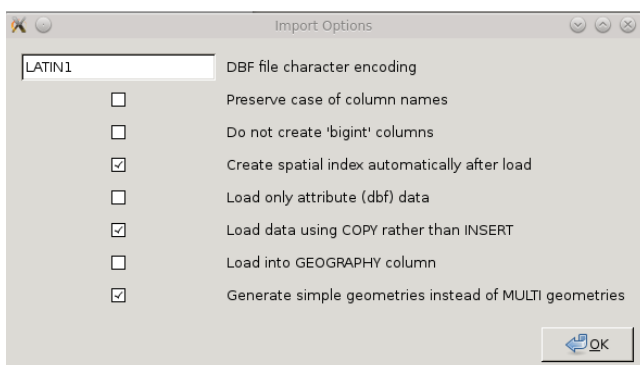


Abbildung 2: Import-Optionen von shp2pgsql-gui

Übung 4

Bei den Optionen setzen Sie das character-encoding auf „LATIN1“, die Checkbox für das Generieren des räumlichen Index, Datenladen mit COPY statt INSERT, und „Generate simple geometries“ (**nur für Flüsse, nicht für die Seen und Gemeinden** – Seen und Gemeinden müssen als Multi-Geometrien geladen werden. Importieren Sie die Seen und Gemeinden daher in einem separatem Import-Schritt mit anderen Optionen). Für das Schema setzen Sie „schweiz“ und für die SRID 21781.

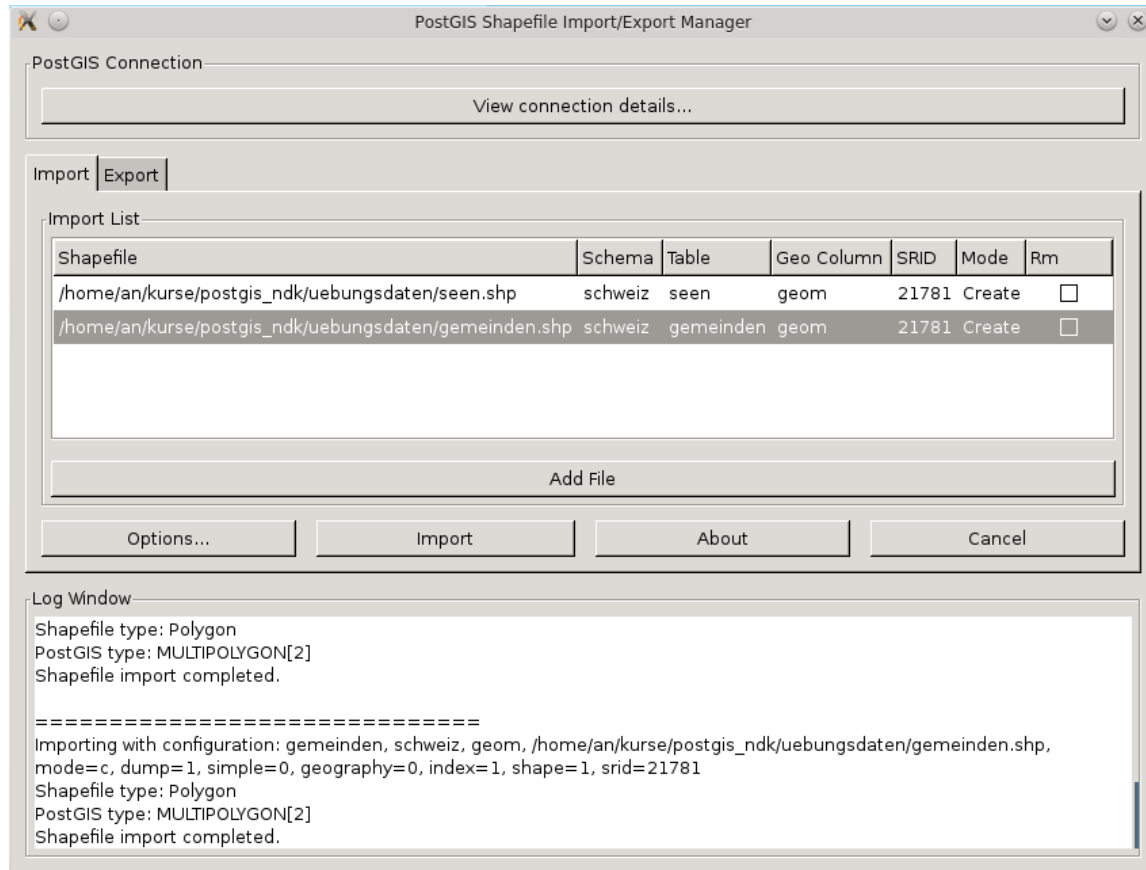


Abbildung 3: shp2pgsql-gui Import-Dialog

Es ist wichtig, dass Sie alle Einstellungen wie SRID, Schema, Encoding, Multi vs. Single Geometry richtig setzen. Ansonsten werden die Lösungen in einigen der nächsten Aufgaben von unseren Musterlösungen abweichen oder nicht funktionieren.

Aufgabe 2 (Laden von CSV Datei):

Sie erstellen eine leere Tabelle mit dem Namen „viertausender_ch“ im Schema „schweiz“ mit den folgenden Spalten und Datentypen. Setzen Sie die Spalte „gid“ als NOT NULL und primary key. Datenquelle: Wikipedia

- gid:integer
- name:text
- hoehe:integer
- xcoor:integer
- ycoor:integer
- rang:integer
- gebirge:text

- schartenhoehe:integer
- schartenname:text
- erstbesteigung:date
- erstbesteiger:text
- webpage:text
- geom:geometry(POINT,21781)

Sie laden die CSV-Datei indem Sie im pgAdmin auf der neu erstellen Tabelle „schweiz.viertausender_ch“ über das Kontext-Menü den Import-Dialog aufrufen. Verwenden Sie die folgenden Einstellungen:

- Format: csv
- Kodierung: UTF-8
- Tab „Spalten“ - Columns to import: alle ausser geom
- Misc Options: „header“ ankreuzen
- Misc Options: Begrenzer “;“

Überprüfen Sie in pgAdmin ob die Daten richtig importiert wurden.

Die Punktgeometrien der Gipfel liegen nun zwar als Integerwerte (getrennt nach Rechtswert und Hochwert) vor, jedoch noch nicht als Postgis-Geometrie. Sie müssen daher noch die Geometriespalte aktualisieren, indem Sie mit Hilfe des Befehls `ST_GeometryFromText()` (siehe http://postgis.org/docs/ST_GeometryFromText.html) und dem SQL UPDATE-Befehl die Spalte „geom“ aus den Spalten „xcoor“ und „ycoor“ erzeugen. Hier ein gültiger WKT (well known text)-String:

```
'POINT(600000 200000)',21781
```

Diesen UPDATE-Befehl müssen Sie zusammensetzen aus Strings und den oben erwähnten Spalten und dem Befehl `ST_GeometryFromText()` als Parameter übergeben.

Aufgabe 3 (Darstellen von PostGIS-Daten in QGIS):

Laden Sie nun in QGIS über den Menüpunkt (Layer → PostGIS-Layer laden) alle Tabellen des Schemas Schweiz. Legen Sie dazu eine neue PostGIS-Verbindung an (siehe Dialog unten):

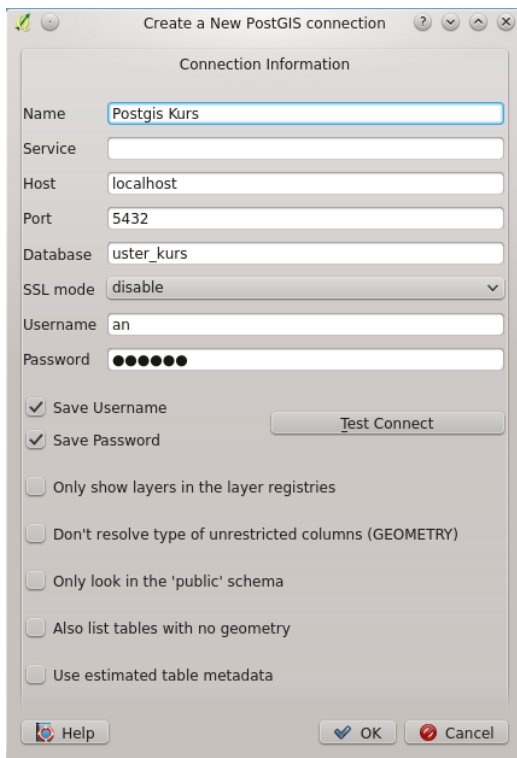


Abbildung 4: PostGIS-Verbindungsdialog in QGIS

Legen Sie eine sinnvolle Ebenenreihenfolge fest und eine einfache Farbgebung (z.B. Seen in blau).
Überprüfen Sie in QGIS auch die Attributtabellen.

Aufgabe 4:

Sie können PostGIS-Daten in QGIS auf mehrere Arten filtern:

1. Über den Layereigenschaftsdialog, Tab „Allgemein“ → „Objektuntermenge“/„Abfrageerstellung“. Bei dieser Methode werden alle Daten in QGIS geladen. Die Filterung erfolgt clientseitig.
2. Für die Darstellung, mit Hilfe des „rule-based Renderers“, indem für die Darstellung verschiedene Regeln erstellt werden mit Subsets
3. Über den DBManager (Menü „Datenbank“ → „DBManager“) - die Filterung erfolgt dabei Server/Datenbank-seitig

Wir probieren die Methoden 1 und 3 aus.

- a) Filtern Sie die 4000er Gipfel und zeigen Sie nur diejenigen Gipfel an, die eine Seehöhe von über 4500m haben. Verwenden Sie dazu die Methode 1.

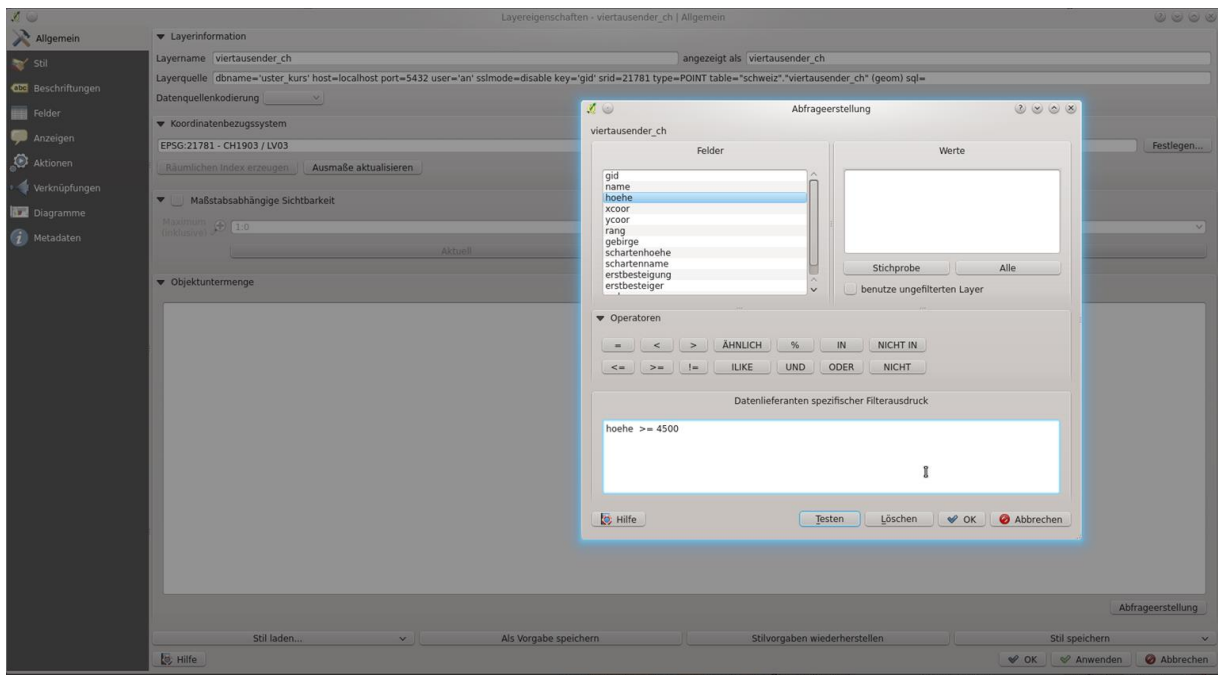


Abbildung 5: Layereigenschaften- und Filterdialog in QGIS

b) Filtern Sie alle Gemeinden nach der Kantonsnummer 23 (Kanton Wallis) über die Methode 3 (DBManager). Navigieren Sie dazu im DBManager zur vorher definierten Postgis-Verbindung, wählen Sie die Tabelle „gemeinden“. Öffnen Sie danach unter dem Menü „Datenbank“ → „SQL Fenster“ den SQL-Dialog und geben Sie die unten im Dialog angegebene Abfrage ein. Testen Sie diese mit „Ausführen (F5)“. Laden Sie dieses Abfrage-Resultat in QGIS indem Sie die Checkbox „Als neuen Layer laden“ aktivieren und die Primärschlüsselspalte (gid) und Geometriespalte (geom) auswählen.

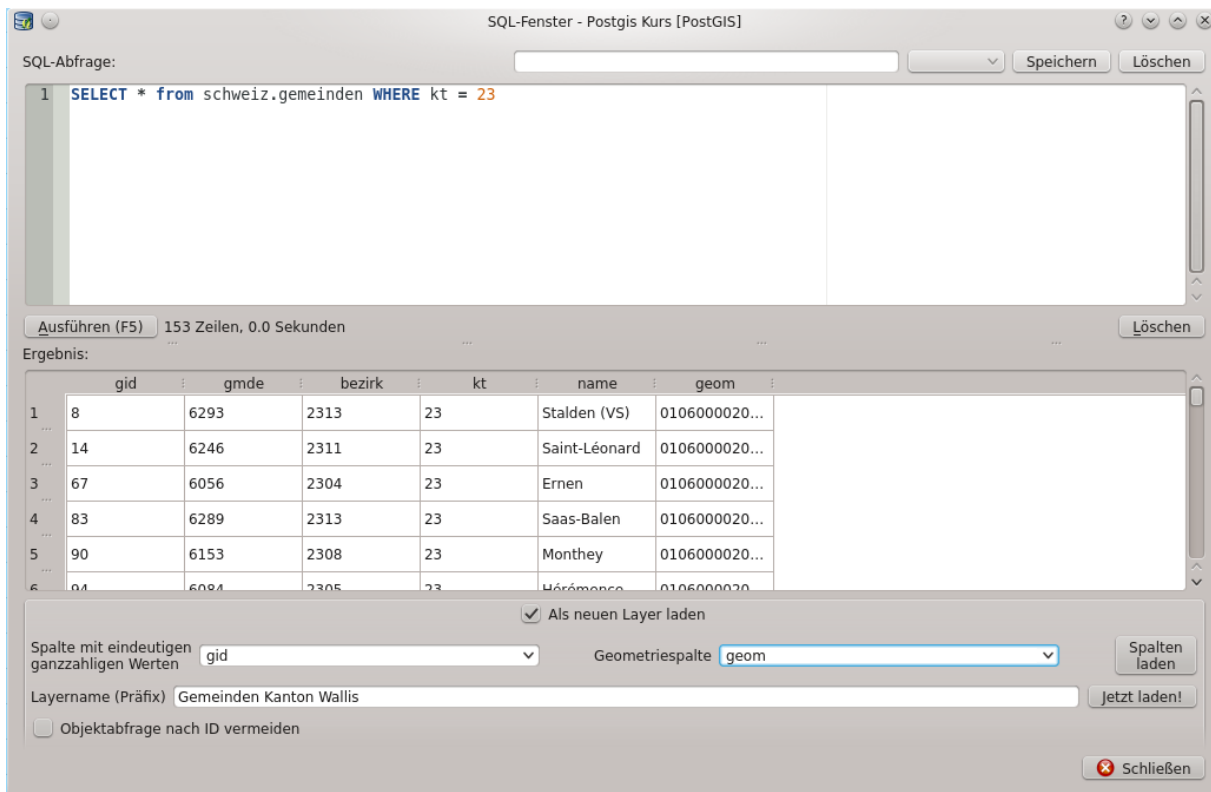


Abbildung 6: SQL-Fenster im QGIS DB-Manager

Beachten Sie, dass nach der Abfrage kein Semicolon (;) folgen soll – sonst kann die Abfrage in QGIS nicht geladen werden. Bei dieser 3. Methode können auch Kombinationen aus mehreren Tabellen (ähnlich views) geladen werden, ausserdem können räumliche Verbindungen zwischen verschiedenen Tabellen verwendet werden.

Aufgabe 5

Zeigen sie den höchsten 4000er der Schweiz auf <http://geojson.io>

Lösungen

Aufgabe 2:

Teil 1 (Tabelle erzeugen):

```
CREATE TABLE schweiz.viertausender_ch
(
  gid integer NOT NULL,
  name text,
  hoehe integer,
  xcoor integer,
  ycoor integer,
  rang integer,
  gebirge text,
  schartenhoehe integer,
  schartenname text,
  erstbesteigung date,
  erstbesteiger text,
  webpage text,
  geom geometry(Point,21781),
  CONSTRAINT viertausender_ch_pkey PRIMARY KEY (gid)
);
```

Teil 2 (Import CSV, interaktiv im GUI mit "import")

Teil 3 (Update Geometrie):

```
UPDATE schweiz.viertausender_ch
  SET geom=ST_GeometryFromText('POINT('||xcoor::text||' '||ycoor||')',21781);
```

Aufgabe 5:

```
select st_asgeojson(st_transform(geom, 4326)) from schweiz.viertausender_ch
where hoehe >= 4500
order by hoehe limit 10
```