

Anleitung für die Installation einer einfachen GIS-Anwendung

Basierend auf:

Ubuntu-Linux
PostgreSQL/PostGIS-Spatial-Datenbank
Quantum GIS

Autor:

Dipl.-Ing. Frank Jäger
Bad Salzuflen

02. Januar 2010

Version 1.0.5

Anleitung für die Installation einer einfachen GIS¹-Anwendung

Ziel:	Aufbau einer GDI ² für die Erfassung und Darstellung eines eigenen georeferenzierten Datenbestandes.
Betriebssystem:	Ubuntu 9.10 ' <i>Karmic Koala</i> '.
Komponenten:	PostgreSQL 8.3 / PostGIS-Datenbank Quantum GIS ³ 1.3 ' <i>Mimas</i> '
Zielgruppe:	GIS-Anfänger mit Linux- und SQL-Kenntnissen

Inhaltsverzeichnis

Anleitung für die Installation einer einfachen GIS-Anwendung.....	2
Installation.....	4
Konfiguration.....	5
Einrichten PostgreSQL.....	5
pg_hba.conf.....	5
postgresql.conf.....	6
User.....	6
Einrichten PostGIS.....	7
Datenbank-Vorlage.....	7
Neue Datenbank.....	7
Tabellen und Felder.....	8
OID.....	8
Geometrie-Feld.....	8
Geometrischer Index.....	9
Einrichten pgAdmin III.....	9
Testdaten.....	10
Daten.....	10
Einrichten Quantum GIS.....	11
Erfassen mit Quantum GIS.....	12
Hintergrundkarte.....	12
Weltkarte.....	12
Deutschlandkarte.....	12
Einstellungen.....	12
Erfassung.....	12
Anlagen.....	13
EPSG-Codes.....	13
Geometrie-Typen.....	13

1 GIS = Geografisches Informations-System

2 GDI = Geodaten-Infrastruktur

3 Quantum GIS siehe <http://www.qgis.org>

Darstellung in dieser Anleitung:

Blau:
Shell-Kommandos auf der Linux-Konsole
oder in Scripten.

Gelb:
SQL-Befehle.
Können wahlweise mit pgAdmin oder im Kommando psql verarbeitet werden.

Installation

In diesem Beispiel werden alle Komponenten auf *einer* Maschine installiert (Server + Client). Es wäre auch möglich, die Komponenten der GDI auf mehrere Maschinen in einem Netzwerk zu verteilen.

Die Datenbank kann z.B. alternativ als Dienst auf einer anderen Hardware im Netz installiert werden oder auf einer virtuellen Maschine. Der Datenbank-Server benötigt dann keine grafische Oberfläche (Linux-Basissystem).

<i>Paket</i>	<i>Beschreibung</i>	<i>Maschine</i>
PostgreSQL	Datenbank-Server	Server
PostGIS	Spatial-Zusatz für PostgreSQL	Server
pgAdmin III	Datenbank-Client	Client
Quantum GIS	Desktop-GIS-Anwendung	Client

Die Paketquellen in '/etc/apt/sources.list' erweitern wir um die Zeilen:

```
# Ubuntu-GIS siehe https://wiki.ubuntu.com/UbuntuGIS
deb http://ppa.launchpad.net/ubuntugis/ubuntugis-unstable/ubuntu karmic main
deb-src http://ppa.launchpad.net/ubuntugis/ubuntugis-unstable/ubuntu karmic main
```

Dann werden werden die benötigten Pakete installiert mit:

```
#!/bin/bash
# My 1st GIS HowTo - Installation der GIS-Komponenten
sudo aptitude update
#
# Server-Komponenten
sudo aptitude install postgresql-8.3 postgresql-doc-8.3
sudo aptitude install postgresql-client-8.3
sudo aptitude install postgresql-contrib postgresql-8.3-postgis
#
# Client-Komponenten
sudo aptitude install pgadmin3 pgadmin3-data qgis
```

In anderen Distributionen können Paketnamen, Versionsstände oder Pfadangaben abweichen. Die Syntax der Datenbank ist aber prinzipiell gleich.

Konfiguration

Einrichten PostgreSQL

pg_hba.conf

Wer von wo auf welche Datenbanken zugreifen darf, ist geregelt in der Datei **pg_hba.conf**.

Wir bearbeiten die Datei mit *gedit* oder in einem anderen Text-Editor. Aufruf:

```
sudo gedit /etc/postgresql/8.3/main/pg_hba.conf
```

Einfügen bzw. ändern der folgenden Zeilen im unteren Teil der Datei:

```
# Database administrative login by UNIX sockets
#local all postgres ident sameuser
local all postgres trust
# TYPE DATABASE USER CIDR-ADDRESS METHOD
# "local" is for Unix domain socket connections only
local all qgis password
local all myUserId ident sameuser
# local all ident sameuser
# IPv4 local connections:
host all all 127.0.0.1/32 md5
host all all 192.168.0.0/16 md5
# IPv6 local connections:
host all all ::1/128 md5
```

... erlaubt folgende Zugriffe:

1. vom lokalen Unix-Socket (Shell) als Datenbank-Superuser 'postgres',
auch ohne Passwort! (trust)
2. vom lokalen Unix-Socket als User 'qgis'
mit Passwortabfrage
3. vom lokalen Unix-Socket als User '*myUserId*',
wenn *myUserId* auch im Linux-System angemeldet ist
(hier die eigene Benutzerkennung einsetzen)
4. als beliebiger Datenbank-User über IP-Port vom gleichen Rechner (localhost)
(zum Beispiel mit dem Programm pgAdmin III)
verschlüsselte Passwort-Abfrage
5. als beliebiger User über IP-Port aus dem *Intranet*
(die IP-Adresse muss beginnen mit '192.168.').

Die erste Zeile (Superuser ohne Passwort) sollte wieder auskommentiert werden, sobald weitere Benutzer eingerichtet wurden, mit denen man arbeiten kann.

Weitere Beispiele in den Kommentaren der Datei und in der PostgreSQL-Dokumentation oder der pgAdmin-Hilfe.

postgresql.conf

Nach der Installation akzeptiert PostgreSQL nur lokale Verbindungen. Um über TCP/IP auf den Datenbank-Service zuzugreifen muss die Konfigurationsdatei angepasst werden:

```
sudo gedit /etc/postgresql/8.3/main/postgresql.conf
```

Anpassung:

```
# - Connection Settings -  
# listen_addresses = 'localhost'  
listen_addresses = '*'
```

.. oder eine Liste der zulässigen Adressen, die auf die Datenbanken zugreifen dürfen.

Damit die Änderungen wirksam werden, muss der Datenbank-Dienst neu gestartet werden:

```
sudo /etc/init.d/postgresql-8.3 restart
```

User

Nun kann als Superuser *postgres* eine neue Benutzerkennung eingerichtet werden.

Der Benutzer *qgis* soll vom Programm 'Quantum GIS' verwendet werden.

Shell im interaktiven Modus:

```
psql -U postgres
```

```
CREATE ROLE qgis LOGIN ENCRYPTED PASSWORD 'geheim' NOINHERIT CREATEDB  
VALID UNTIL 'infinity';
```

```
\q
```

Der Befehl `psql ohne` den Parameter `-f` startet den interaktiven Modus.

Es kann ein SQL-Kommando eingegeben werden. Das Kommando muss mit `;` (Semikolon) beendet werden.

Der interaktive SQL-Modus wird mit `\q` wieder verlassen.

Nun als User *qgis* (-U) eine gleichnamige Wartungs-Datenbank für den Owner *qgis* (-O) anlegen:

```
createdb -U qgis -O qgis qgis;
```

Einrichten PostGIS

Datenbank-Vorlage

Die PostGIS-Spracherweiterung wird nun in einer speziellen Datenbank-Vorlage (*Template*) eingerichtet. Diese Vorlage wird immer verwendet, wenn eine neue Datenbank mit Spatial-Funktionen benötigt wird.

```
createdb -p 5432 -U postgres          template_postgis
createlang -p 5432 -U postgres plpgsql template_postgis
cd /usr/share/postgresql-8.3-postgis
psql      -p 5432 -U postgres -d template_postgis -f lwpostgis.sql
psql      -p 5432 -U postgres -d template_postgis -f spatial_ref_sys.sql
```

createdb: Eine neue Datenbank wird angelegt

createlang: eine neue Sprache wird angelegt (hier die PostGIS-Funktionen)

/usr/share/ ..: Installationsort der SQL-Dateien für *Ubuntu*.
Bei anderen Distributionen eventuell abweichend.

lwpostgis.sql: Definieren der PostGIS-Funktionen

spatial_ref_sys.sql: Eine Referenztabelle mit Koordinatensystemen wird angelegt

Für die weiteren Schritte wird hier exemplarisch der neue User 'qgis' benutzt, der das Recht besitzt, neue Datenbanken anzulegen.

Neue Datenbank

Auf der Basis dieser Vorlage kann nun eine neue Spatial-Datenbank 'mygisdb' angelegt werden.

Mit Dienstprogramm:

```
createdb -U postgres -O qgis -E UTF8 -T template_postgis mygisdb
```

.. oder mit SQL:

```
psql -U postgres
```

```
CREATE DATABASE mygisdb
  WITH OWNER = qgis
      ENCODING = 'UTF8'
      TEMPLATE = template_postgis
      TABLESPACE = pg_default;
```

```
\q
```

Dann muss der User qgis noch berechtigt werden, in der Datenbank qgis Metadaten einzutragen:

```
psql -U postgres -d mygisdb
```

```
GRANT ALL ON TABLE geometry_columns TO qgis;
GRANT ALL ON TABLE spatial_ref_sys TO qgis;
```

```
\q
```

Die Tabelle *spatial_ref_sys* enthält die Parameter der Koordinatensysteme.
Die Tabelle *geometry_columns* speichert Hinweise auf alle Tabellen der Datenbank, die Geometriefelder enthalten.

Tabellen und Felder

Nun fehlt noch die Datenstruktur in der neuen Datenbank.

```
psql -U qgis -d mygisdb
```

```
create table poi (  
            id          serial NOT NULL,  
            poi_typ    int4,  
            anzeige    varchar(24),  
            CONSTRAINT pk PRIMARY KEY (id)  
        )  
WITH OIDS;
```

```
\q
```

OID

Der Parameter 'WITH OIDS' ist eine Spezialität von PostgreSQL. Jede Datenbankzeile erhält dadurch eine eindeutige Kennung mit Namen 'OID' (Long Integer). Diese Kennung ist nützlich um bei Änderungen an den Daten die Zeile eindeutig zu adressieren. Frühe Versionen des Programms *Quantum GIS* benutzten als Standard dies Feld *OID* und akzeptierten keinen anderen eindeutigen Primärindex.

Dieser Index ist jedoch *nicht langfristig stabil*. Er sollte nicht für die Verknüpfung von Tabellen (*Foreign Key*) verwendet werden. Bei der Sicherung der Datenbank als SQL-Dump wird die *OID nicht* standardmäßig ausgegeben. Beim Laden wird die *OID* dann neu gebildet.

Wenn für die Verknüpfung von Tabellen kein fachlich eindeutiges Feld zur Verfügung stehen, kann ein Feld vom Typ *'serial'* verwendet werden, das im Gegensatz zur *OID* auch langfristig Bestand hat.

Die Tabelle in diesem Beispiel hat mit *id* (Serial) und *OID* also zwei Schlüssel.

Geometrie-Feld

Das Datenbankfeld für die Koordinate wird mit der Funktion *AddGeometryColumn* angelegt. Das stellt sicher, dass einige Constraints angelegt werden und ein Eintrag in der Metadatatabelle *'geometry_columns'* auf dieses Feld hinweist. Die Constraints (Einschränkungen) überwachen das korrekte Koordinatensystem, den Geometrietyp und die Dimensionen.

```
psql -U qgis -d mygisdb
```

```
select addgeometrycolumn('poi','the_geom',4326,'POINT',2);
```

```
\q
```

Dieser SQL-Befehl kann interaktiv zwischen *'psql'* und *'\q'* eingegeben werden, oder vom Client über das Programm *pgAdmin*.

Parameter:

- 'poi' = Name der Tabelle
- 'the_geom' = Name des Feldes
- 4326 = SRID⁴ des Koordinatensystems, siehe Anlage EPSG-Codes
- 'POINT' = Punkt-Geometrie, siehe Anlage Geometrie-Typen
- 2 = Anzahl der Dimensionen (hier ebene Koordinaten, ohne Höhe)

Es wird hier das geografische Koordinatensystem verwendet; also Länge und Breite in Grad, EPSG 4326. Das ist am einfachsten, wenn *weltweit* Koordinaten erfasst werden sollen. Daraus können z.B. ganz einfach Links für GoogleMaps oder [OpenStreetMap](#) abgeleitet werden.

Allerdings hat das auch einige Nachteile. Im Bereich von Deutschland erscheint eine Karte seitlich gestreckt. Bei einigen Programmen funktioniert die Berechnung des Maßstabes nicht und man kann nicht so einfach Strecken aus den Koordinaten berechnen.

Wenn nur in einem regionalen Bereich Daten erfasst werden sollen, kann statt dessen das Koordinatensystem *Gauß-Krüger* oder *UTM/ETRS89* verwendet werden..

Geometrischer Index

Für die schnelle *geometrische* Suche (z.B. Präsentation eines Ausschnitts in einem GIS) sollte das Geometrie-Feld indiziert werden. Für ein Geometriefeld eignet sich nicht die Index-Methode, die für alphanumerische Felder verwendet wird (*btree*). Es gibt dafür die Methode '*gist*'.

SQL-Befehl:

```
CREATE INDEX gist_poi
ON poi
USING gist (the_geom);
```

Einrichten pgAdmin III

Sobald der Datenbank-Dienst auch Verbindungen über einen IP-Port zulässt, können SQL-Kommandos auch mit dem Client *pgAdmin* ausgeführt werden.

Alternative von einem Client-Rechner:

SSH-Verbindung über PuTTY und *semi-lokale* SQL-Ausführung mit 'psql'.

pgAdmin-Menü: 'Datei' / 'Server hinzufügen', dann Ausfüllen:

Adresse:	<i>localhost</i>	Rechnername oder IP
Beschreibung	<i>qgis-local</i>	angezeigter Name für diese DB-Verbindung
Port:	<i>5432</i>	der Standard-Port des Dienstes
Wartungs-DB:	<i>qgis</i>	Es hat sich als nützlich erweisen, für jeden Benutzer eine eigene (leere) Datenbank anzulegen.
Benutzername:	<i>qgis</i>	

4 SRID = Spatial-Reference ID = Kennung des Koordinatensystems

Testdaten

Bevor etwas mit QGIS angezeigt werden kann, muss etwas in die Tabelle eingetragen werden:

```
psql -U qgis -d mygisdb
```

```
INSERT INTO poi (poi_typ, anzeige, the_geom)
VALUES (
    1,
    'Ein Punkt',
    GeometryFromText('POINT(9.4432 52.652)', 4326)
);
INSERT INTO poi (poi_typ, anzeige, the_geom)
VALUES (
    2,
    'Noch ein Punkt',
    GeometryFromText('POINT(9.2221 51.999)', 4326)
);
```

```
\q
```

Alternativ kann die SQL mit pgAdmin verarbeitet werden:

- Doppelklick auf den eingerichteten Server um die Verbindung zu öffnen.
- Öffnen: Datenbanken / mygisdb / Schemata / public
- das SQL-Icon im Kopf öffnet ein neues Fenster
- SQL in der oberen Hälfte hinein kopieren
- den grünen Pfeil zum Ausführen drücken

Daten

Das Feld 'poi_typ' enthält die Werte 1 und 2 und soll das anzuzeigende *Symbol* steuern.

Der Inhalt des Feldes 'anzeige' soll als *Label* in der Karte angezeigt werden.

Mit der Funktion *GeometryFromText* wird die lesbare Koordinate in das binär gepackte Geometriefeld eingetragen.

Die Funktionsparameter (WKT⁵) enthalten den Geometrietyp, die Koordinate und das Koordinatensystem. Die Dezimalstelle ist mit '.' (Punkt) formatiert.

Latitude und Longitude (Rechtswert und Hochwert, x und y) sind mit einer Leerstelle (Blank) getrennt.

Die Koordinatenwerte müssen zu dem angegebenen System passen, hier geografische Koordinaten.

5 WKT = Well Known Text = Eingabeformat für Geometriefelder

Einrichten Quantum GIS

Menü: Layer (Ebene) / Füge eine PostGIS-Ebene hinzu / PostgreSQL-Verbindungen / neu

Eingaben:

Name = mygis_poi

Host = localhost

Datenbank = mygisdb

Port = 5432

Benutzername = qgis

Passwort = geheim

Speichere Passwort = ja

Verbinden / Markieren: 'public'. 'poi' (the_geom) / Hinzufügen

Nun werden die zwei Punkte am Rand des Kartenbereiches angezeigt.

Rechtsklick auf Layer / poi / Eigenschaften:

Register: **Darstellung**

Legendentyp:

eindeutiger Wert

Klassifizierungsfeld:

'poi_typ'

Schaltfläche:

Klassifizieren

Dann die Werte '1' und '2' nacheinander markieren, Symbole aus der Liste dazu auswählen und die Größe auf ca. 10 erhöhen.

Register: **Beschriftungen**

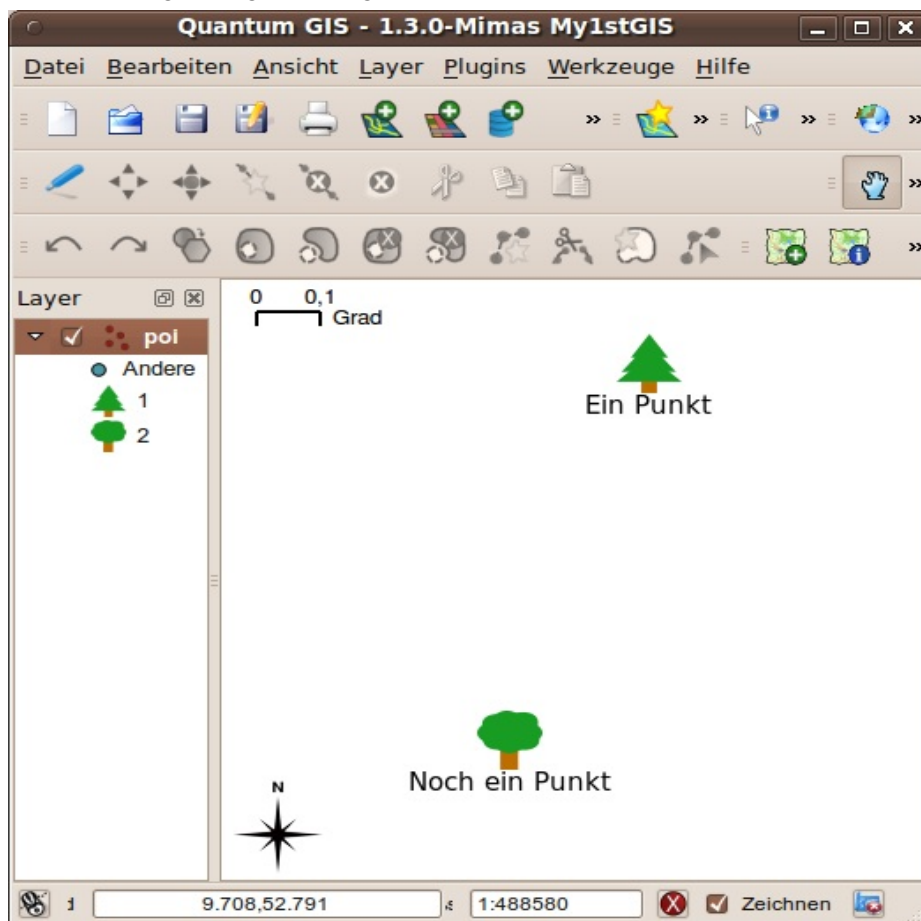
Zeige Beschriftungen an:

ja (Haken setzen)

Beschreibungsfeld:

'anzeige' (Auswahl)

Nach ein paar Feineinstellungen zeigt sich folgende Karte:



Erfassen mit Quantum GIS

Weitere Punkte sollen nicht *numerisch* (als Koordinate) sondern *grafisch* erfasst werden.
Dazu wird die Datenbank änderbar gemacht. Für die Orientierung benötigen wir eine Hintergrundkarte.

Hintergrundkarte

Weltkarte

Menü 'Layer / WMS-Layer hinzufügen'.

Button 'Neu'

Name = Weltkarte

URL = <http://www2.demis.nl/wms/wms.asp?wms=WorldMap&VERSION=1.1.1&SERVICE=WMS>

Die neue Zeile dann auswählen und Schaltfläche 'Verbinden'.

Alle Layer auswählen und Schaltfläche 'Hinzufügen'.

Deutschlandkarte

Menü 'Layer / WMS-Layer hinzufügen'.

Button 'Neu'

Name = Deutschlandkarte

URL = <http://wms1.ccgis.de/cgi-bin/mapserv?map=/data/umn/germany/germany.map&VERSION=1.1.1&SERVICE=WMS>

Die neue Zeile dann auswählen und Schaltfläche 'Verbinden'.

Alle Layer auswählen und Schaltfläche 'Hinzufügen'.

Einstellungen

Koordinatensystem (falls mehrere verfügbar) ändern auf 'WGS 84'.

Erfassung

Werkzeugleiste '*Digitalisieren*' einschalten (Rechtsklick in den Werkzeug-Bereich).

Rechtsklick auf die Ebene: '*Bearbeitungsstatus umschalten*' (Schaltfläche mit 'Stift' markiert.) .

Schaltfläche 'Punkt digitalisieren', Klick in die Karte.

Ausfüllen: poi_typ (Ganzzahl) = '1', anzeige (Text) = 'Test', Schaltfläche 'OK'.

Schaltfläche "*Bearbeitungsstatus umschalten*", 'Wollen Sie die Änderungen am Layer poi speichern?' = Save

Die Koordinate des Mausclicks ist nun als Punkt-Geometrie in der Datenbank gespeichert.

Anlagen

EPSG-Codes

6

Geografische Koordinaten

4326 Weltkoordinaten Länge/Breite, Einheit: Grad

UTM

Liegenschaftskataster ab ca. 2010

25832 ETRS89 / UTM Zone 32N

Gauß-Krüger,

Ellipsoid *Bessel*

Liegenschaftskataster alte Bundesländer bis ca. 2009, Einheit: Meter

31466 2. Meridianstreifen (um 6° Ost)

31467 3. Meridianstreifen (um 9° Ost)

31468 4. Meridianstreifen (um 12° Ost)

31469 5. Meridianstreifen (um 15° Ost)

Gauß-Krüger,

Ellipsoid *Krasovsky*

Liegenschaftskataster neue Bundesländer, Einheit: Meter

2397 3. Meridianstreifen (um 9° Ost)

2398 4. Meridianstreifen (um 12° Ost)

2399 5. Meridianstreifen (um 15° Ost)

Siehe auch in der Tabelle '*spatial_ref_sys*' in einer PostGIS.-Datenbank.

Geometrie-Typen

Für Spatial-Datenbanken

POINT **Punkt**, einfache Koordinate

LINESTRING **Linie** über mehrere Stützpunkte, ohne Unterbrechung

MULTILINESTRING **Linie** über mehrere Stützpunkte, Unterbrechungen möglich

POLYGON **Fläche**, geschlossener Umring. nur eine Teilfläche

MULTIPOLYGON **Fläche**, die aus mehreren Teilflächen bestehen kann oder Fläche mit Aussparungen (Löcher)

6 EPSG = ehemals '*European Petroleum Survey Group*', siehe www.epsg.org