

- PostGIS - Free Software Geospatial Database

Abschlussveranstaltung
eLearning-Seminar „Geodatenbanken“
30. Juni 2007, Oldenburg

- ▶ **Einführung**
 - **Freie Software**
 - **Historie von PostgreSQL/PostGIS**
 - **Installation (Win32, GNU/Linux)**
- ▶ **PostGIS-Features**
 - **Räumliche Operationen / Räumliche Prädikate**
- ▶ **PostGIS-Kartenklienten**
- ▶ **Ausblick**

▶ **Philosophie:**

- ▶ Zukunftsweisende EDV-Konzepte herstellerunabhängig und auf Basis von Freier Software entwickeln und umsetzen

▶ **Kernkompetenzen:**

- ▶ IT-Dienstleistungen zu strategischer Beratung, Projekt-Management & Umsetzung sowie Geographische Informationssysteme (GIS)

▶ **Engagement:**

- ▶ Linux-Verband und im IuK Netzwerk Osnabrück
- ▶ FSF Europa
- ▶ Gründer der führenden Übersicht für Freie Software im GIS-Bereich (www.freegis.org).

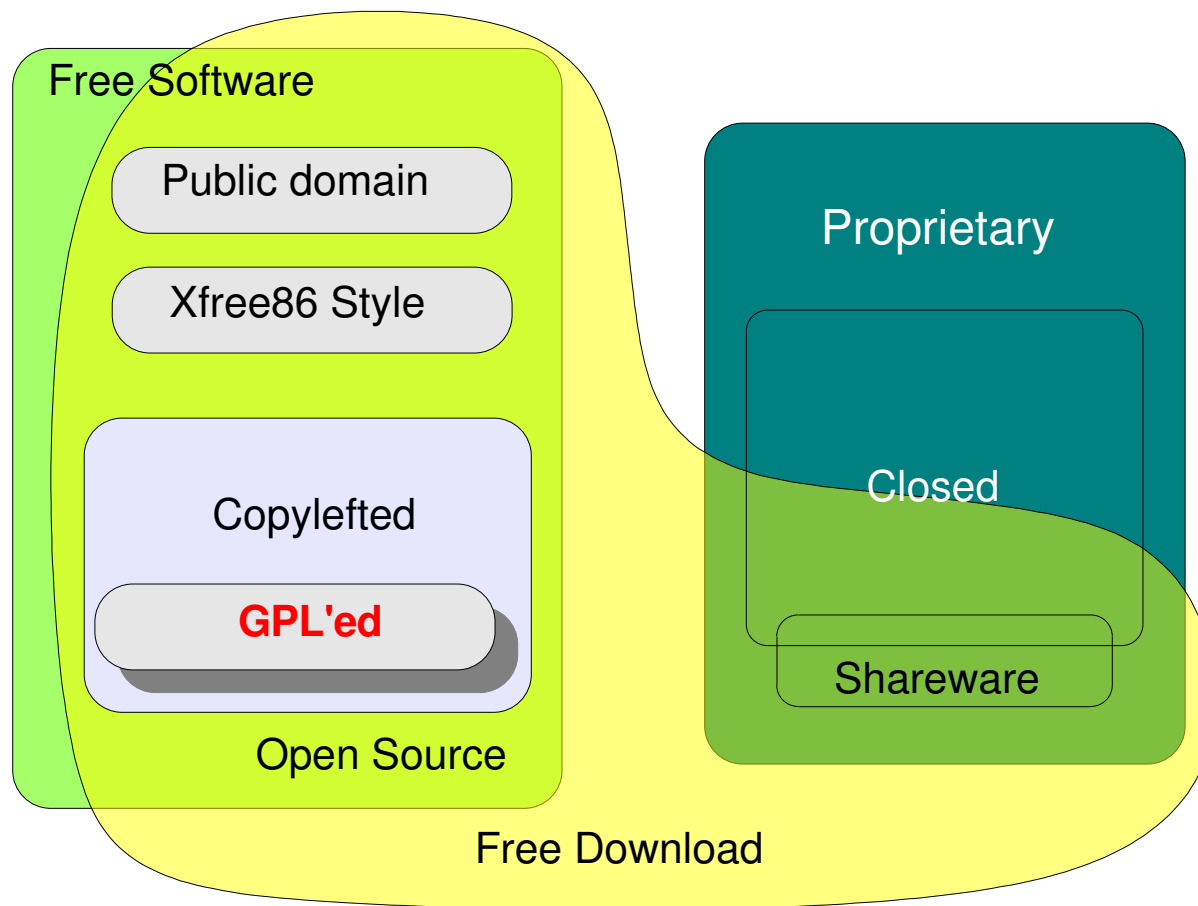
► Was ist Freie Software?

- Die Freiheit, das Programm für jeden Zweck zu nutzen.
- Die Freiheit, das Programm zu verstehen und anzupassen.
- Die Freiheit, das Programm als Kopie zu verbreiten.
- Die Freiheit, das Programm zu verbessern und diese der Öffentlichkeit zur Verfügung zu stellen.

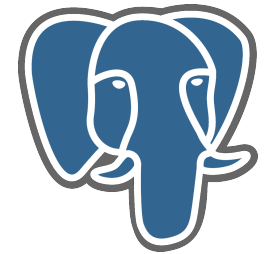


(nach R. Stallman)

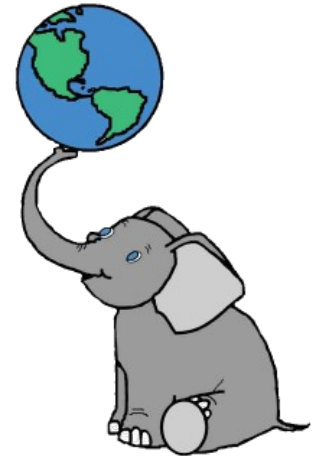
► Welche Freien Lizenzmodelle gibt es?



- 1986-1994: Postgres
 - Forschungsprojekt
- 1995: Postgres95
 - SQL Support
- 1996-2000: PostgreSQL 6.0-7.0
 - Freie Software Projekt
 - Stabilität, Performance, MVCC, Toasted Tuples
- 2001-2005: PostgreSQL 7.0-8.0
 - Schemata, Optimizer, Win32, PGAdmin3
- 2007: PostgreSQL 8.2.4



- 2001
 - SFSQL als Designgrundlage, erste Version
 - Anbindung an UMN MapServer
- 2002
 - Basisfunktionen, Index
- 2003 – 2005
 - GEOS, Lightweight Geometries, GiST, v1.0.0
- 2006
 - OGC Simple Feature for SQL (SFSQL) Standard compliant



- ▶ **Download der Quellen von**
<http://postgis.refractions.net>
- ▶ **Installation unter Windows**
 - PostgreSQL beinhaltet bereits eine PostGIS-Version, jedoch ist diese meist veraltet.
 - separater PostGIS-Installer vorhanden.
- ▶ **Installation unter GNU/Linux**
 - Bsp. Debian: `apt-get install postgresql-postgis`
 - Aus den Quellen: `./configure && make && make install`

► Einspielen der PostGIS-relevanten SQL-Dateien:

```
createuser [DB-Benutzer]
```

```
createdb -O [DB-Benutzer] [Datenbank]
```

```
createlang -U postgres plpgsql [Datenbank]
```

```
psql -U [DB-Benutzer] [Datenbank] < lw_postgis.sql
```

```
psql -U [DB-Benutzer] [Datenbank] < spatial-ref-sys.sql
```

► Dateien importieren

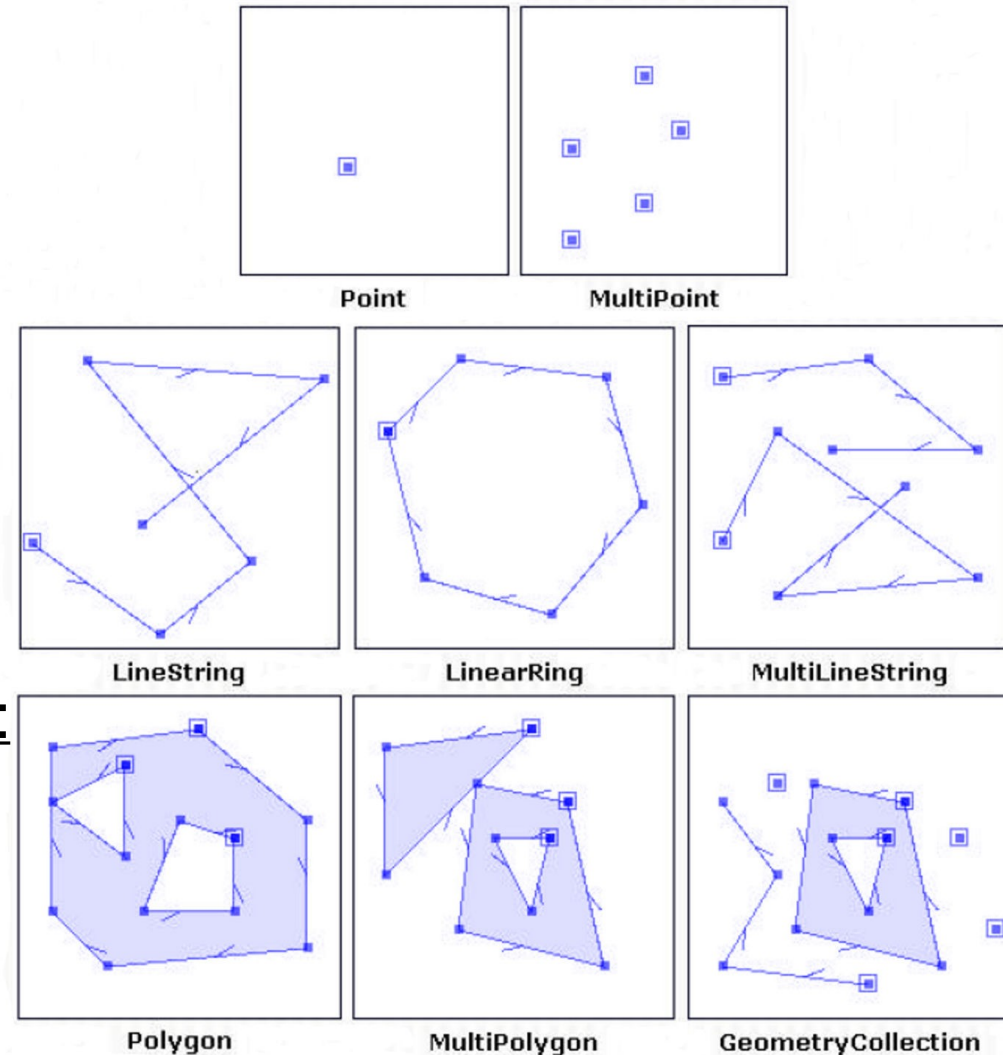
- shp2pgsql
- ogr2ogr
- ...

Simple Feature Model for SQL:

- Point/MultiPoint
- LineString/MultiLineString
- Polygon/MultiPolygon
- GeometryCollection

Repräsentation in der Datenbank:

- WKT (WellKnownText) oder
- WKB (WellKnownBinary)

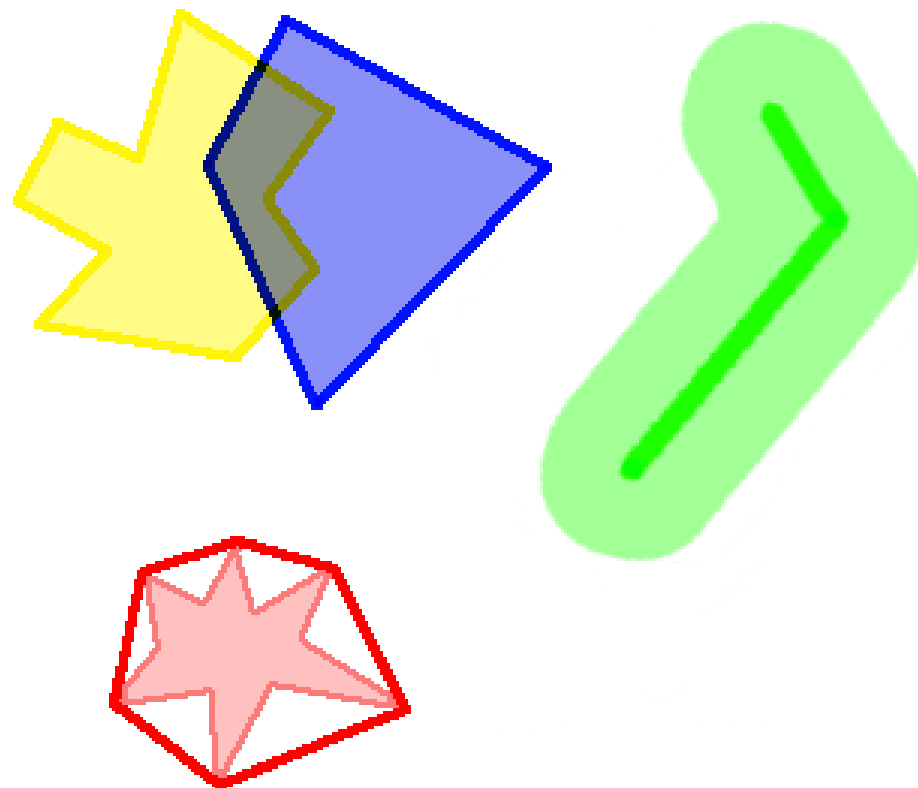


aus: Daluege et al., PostGIS-Tutorial, HCU Hamburg, Geomatik

- Lizenz: GNU GPL
- 2D räumlicher Index (GiST RTree)
- Projektionen (SRS)
- ~300 räumliche Funktionen
 - Flächen, Längen, Entfernungen, ...
- Linear Measurement
- Import / Export
- Long Transaction Locking

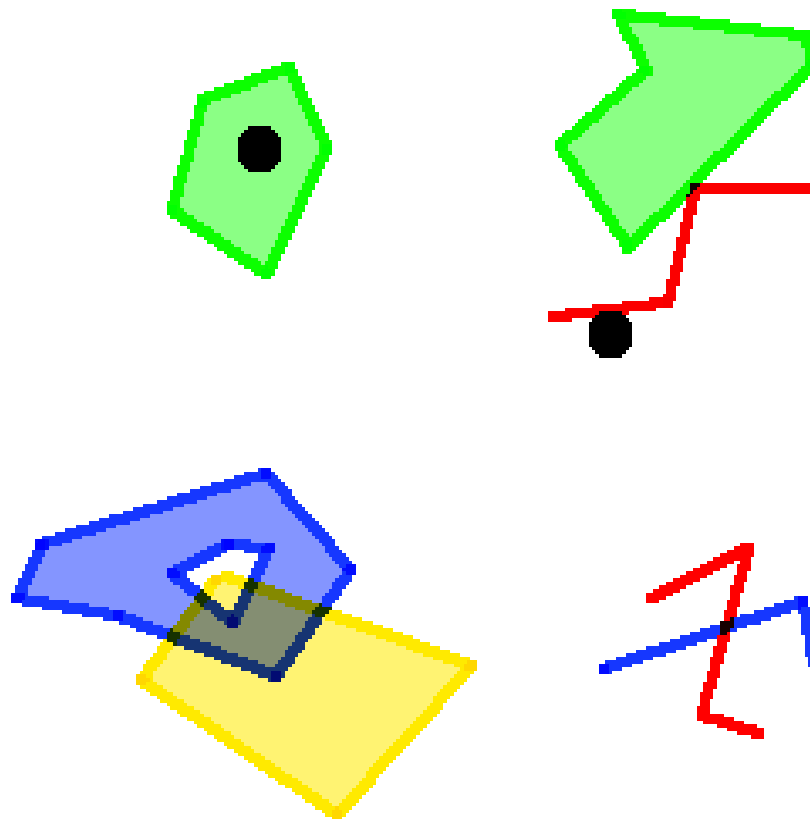
- **Räumliche Analysen**

- Union
- Intersection
- Difference
- Symetric Difference
- Convex Hull
- Buffer
-



• Räumliche Prädikate

- Equals
- Disjoint
- Intersects
- Touches
- Crosses
- Within
- Contains
- Overlaps
-



► SRS (Spatial Reference System)

- PROJ.4-Bibliothek (<http://proj.maptools.org>) integriert
- Projektionen sind durch sog. EPSG-Codes kodiert
- Beispiel: EPSG:4326 entspricht

```
# WGS 84
<4326> +proj=longlat +ellps=WGS84 +datum=WGS84
+no_defs  <>
```
- Koordinaten-Transformation on-the-fly möglich

- ▶ **PGAdmin3 (<http://www.pgadmin.org/>)**
 - PostgreSQL-Frontend
 - GUI zur Datenbankbearbeitung
- ▶ **shp2pgsql/pgsql2shp (bei PostGIS dabei)**
 - Kommandozeilenwerkzeug
 - gut für Batch-Import
- ▶ **ogr2ogr (<http://www.gdal.org/ogr>)**
 - Kommandozeilenwerkzeug
 - kann auch andere Formate lesen/schreiben
- ▶ **gphsql2shp (<http://sdteffen.de/gis/gpgsql2shp.html>)**
 - GUI für shp2pgsql/pgsql2shp

The screenshot shows the PGAdmin 3 interface. On the left is a tree view of the database structure. The main window displays the details of the 'frida_gew' table, including its columns: 'gid', 'gshapeid', 'gnameid', and 'the_geom'. Below this, a SQL console window shows the execution of a query: `SELECT gnameid, astext(the_geom) FROM frida_gew WHERE gnameid = 1`. The results table shows one row: '1 Rubbenbruch-See' with a 'MULTIPOLYGON' geometry.

The 'gpgsql2shp' dialog box is shown. It has a 'Connection' section with fields for Host (localhost), User (test), DataBase (test), and Port (5432). There is a 'Shapefile' section with a 'Name' field and a 'Save in folder' dropdown set to 'devcpp'. Below are 'Options' for Cursor, Binary Geometry Column (set to 'the_geom'), Mode, Identifiers Case (Keep), and Encoding (SQL_ASCII). The 'Output' section has fields for Commandline, Output, and Errors.

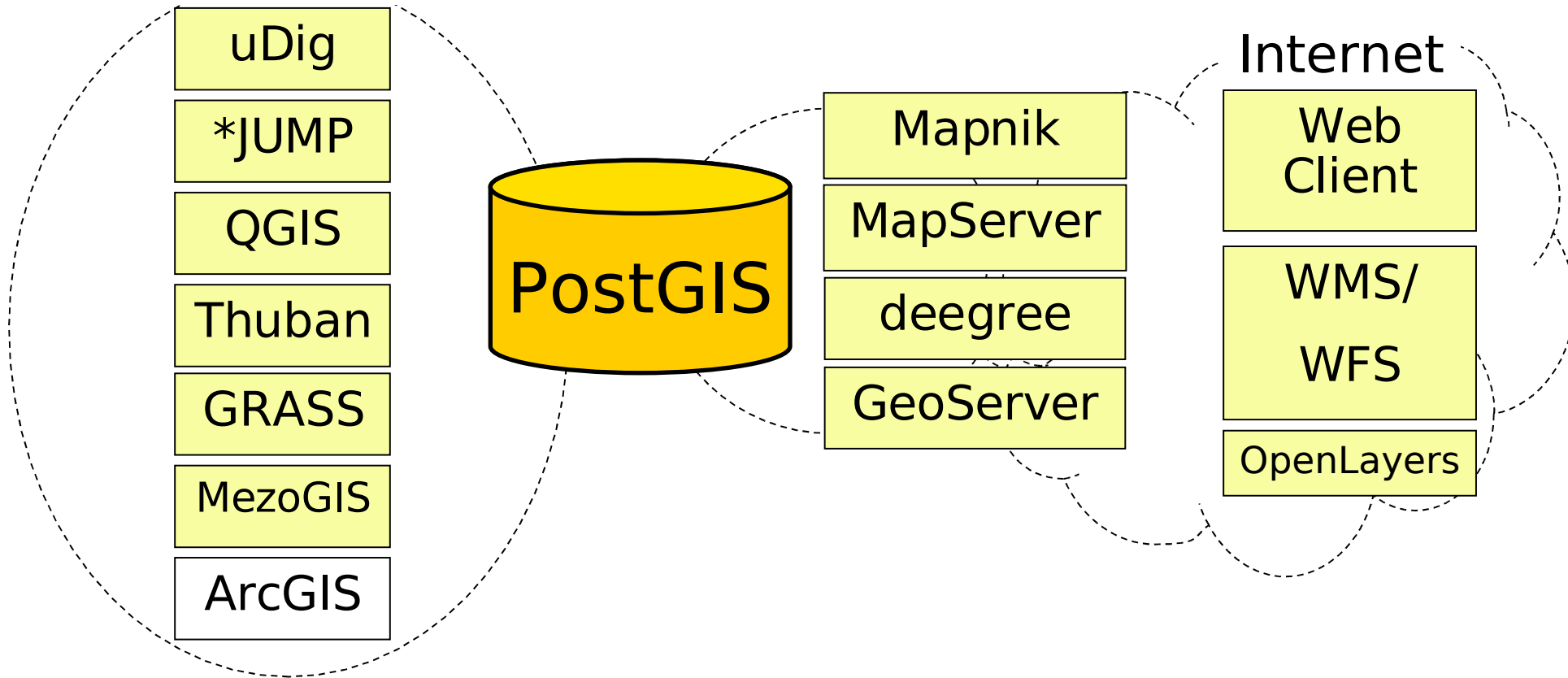
The 'gshp2pgsql' dialog box is shown. It has a 'Connection' section with fields for Host (localhost), User (sdteffen), DataBase (dcms), and Port (5432). There is a 'Shapefile' section with a '(None)' selection. Below are 'Options' for SRID (-1), Geometry Column (the_geom), Mode (Create, populate), Format (Use Dump Format), Index (Create Spatial Index), and Encoding (UTF8). The 'Additional Scripts' section has fields for Pre-load script and Post-load script. The 'Output' section has fields for Commandline, Output, and Errors.

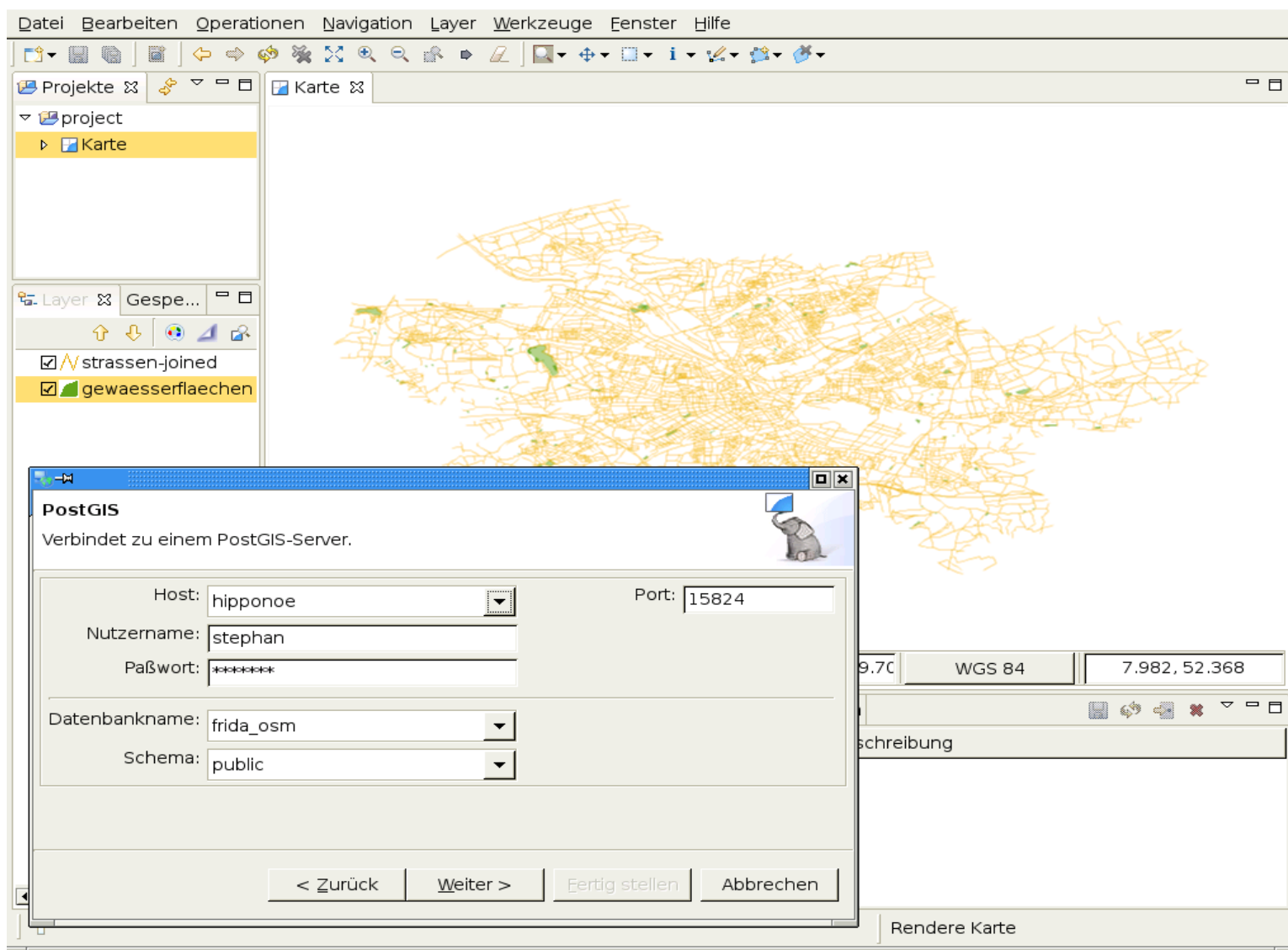
PGAdmin 3 mit SQL-Konsole

gpgsql2shp/gshp2pgsql:
Graphische Frontends

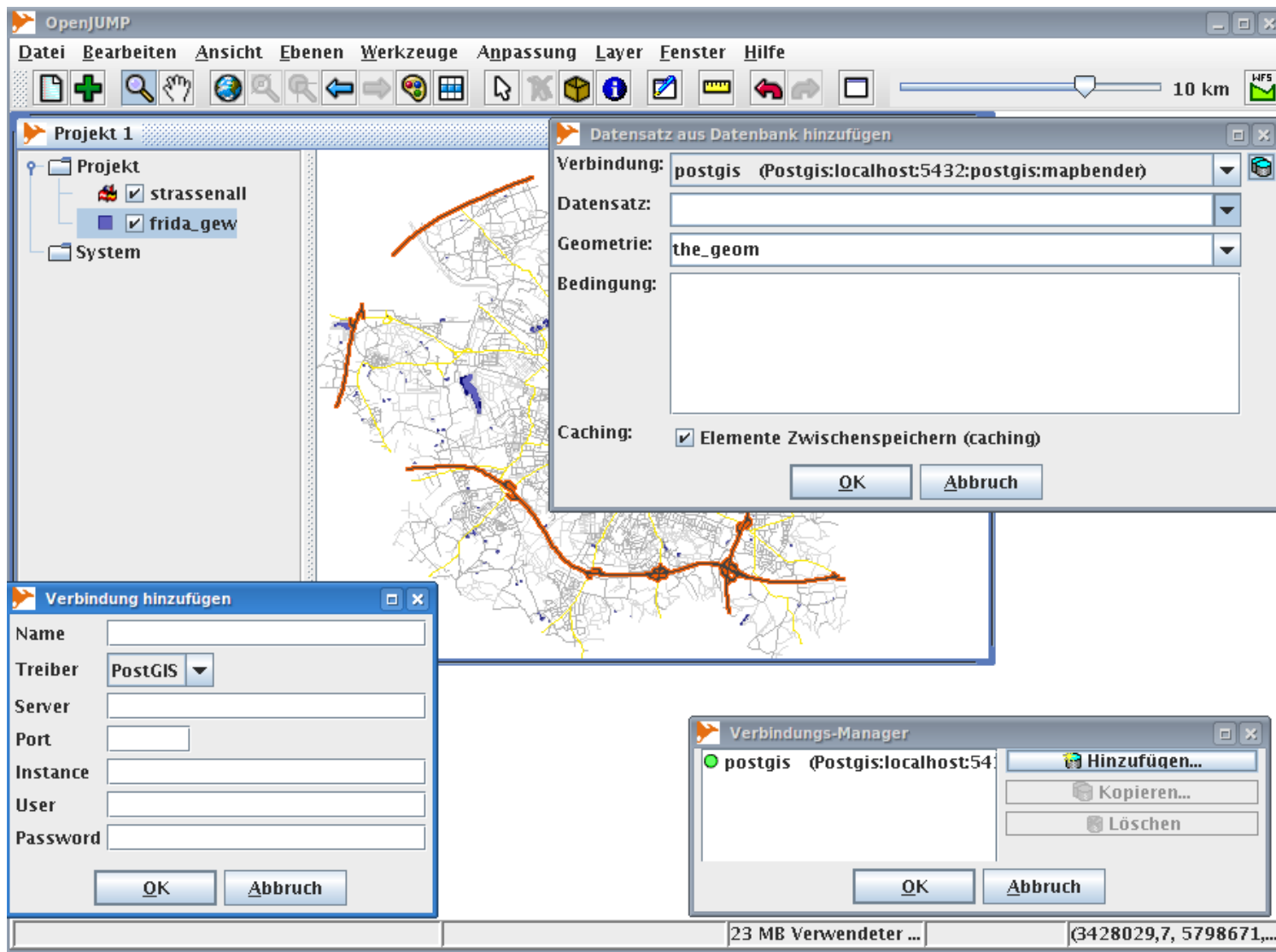
Desktop

Internet





UDIG (<http://udig.refractions.net>)



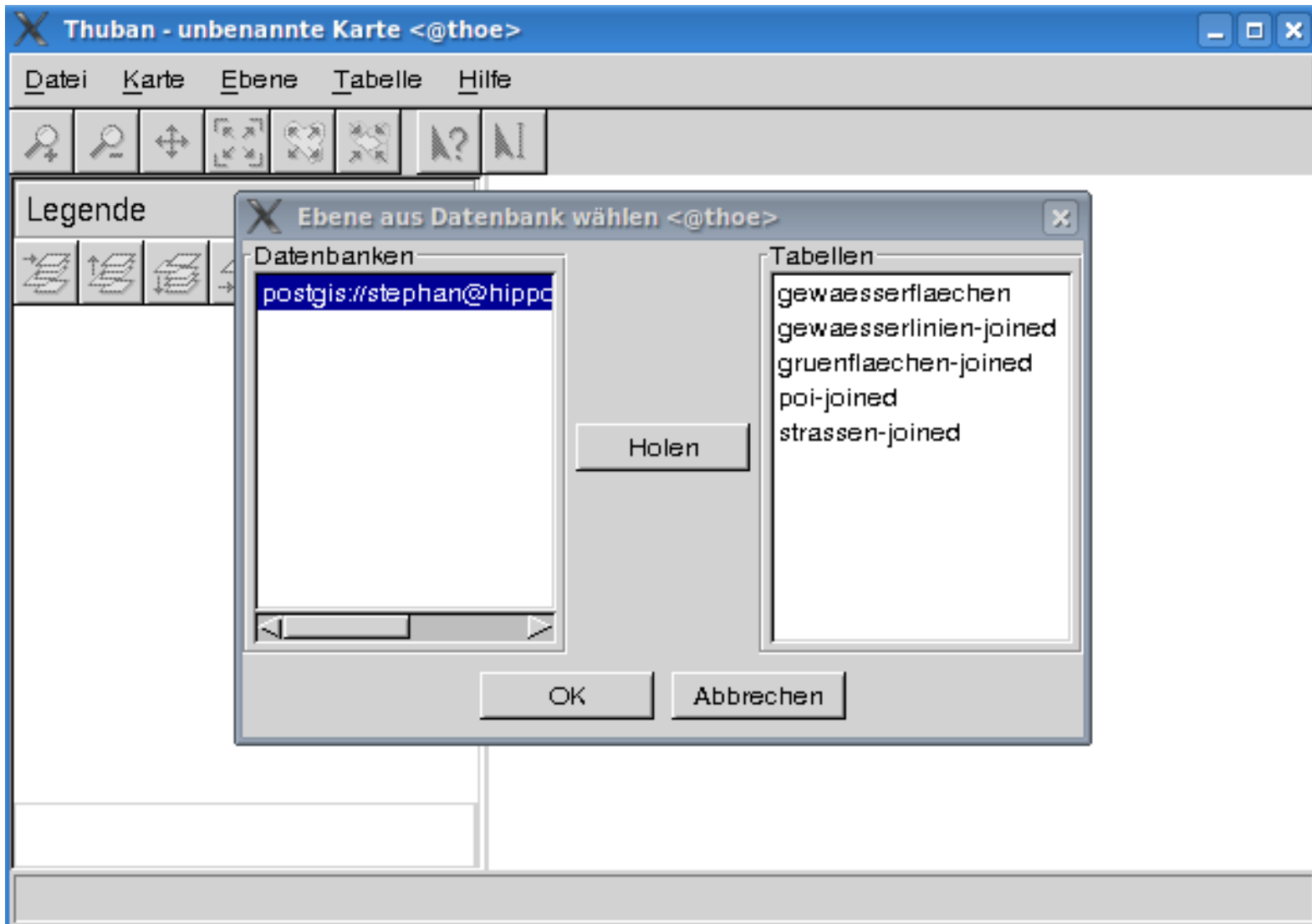
OpenJUMP (<http://openjump.org>)

The screenshot shows the QGIS 0.9.0-Unstable interface. The main window displays a map of a city area with various features. A dialog box titled "PostGIS Tabelle(n) hinzufügen" is open, showing the "PostgreSQL Verbindungen" section with "postgis-localhost" selected. Below this, the "Tabellen:" section contains a table with one entry:

Typ	Name	Sql
	"public"."frida_gew" (the_geom)	

The dialog also shows the "Kodierung:" section set to "ISO-8859-15".

QGIS (<http://www.qgis.org>)



Thuban (<http://www.thuban.intevation.org>)

The screenshot displays the MezoGIS 0.1.4 application interface. On the left, a map window titled 'Map 1' shows a map of Europe with several countries highlighted in yellow, one in blue, and one in red. The top toolbar includes 'Zoom In', 'Best Fit', 'Refresh', and 'Save As'. Below the map, a status bar shows coordinates: 4318091.62306x and 539251.911848y.

In the center, a 'Query Editor' window contains the SQL query: `select * from europe where code = 'FR'`. The 'SQL' tab is active.

On the right, a 'mezoGIS 0.1.4' window shows a 'Maps and Layers' panel with 'Layer 2', 'Layer 3', and 'Layer 1'. Below it, a 'Project' window displays a table of data:

code	name	pop_cntry	curr_type	curr_code	fips	t
3	Belgium	3416945	Lek	ALL	AL	C
4	Bulgaria	55335	Peseta	ADP	AN	C
5	Denmark	7755406	Schilling	ATS	AU	C
6	Estonia	10032460	Franc	BEF	BE	C
7	Bosnia and Herzegovina	2656240			BK	C
8	Bulgaria	8943258	Lev	BFL	BU	C
9	Denmark	4667750	Danish Krone	DKK	DA	C
10	Ireland	5015975	Pound	IEP	EI	OGM
11	Estonia	1590808	Kroon	EKR	EN	OGM
12	Czech Republic	10321120	Koruna	CSK	EZ	OGM
13	Finland	5031379	Markka	FIM	FI	OGM
14	Faroe Islands	47067	Danish Krone	DKK	FO	OGM

The bottom status bar shows: 'mezoGIS 0.1.4 | Map 1 | Layer 1: SELECT | Query Editor | Layer Properties | 17:37'.

MezoGIS (<http://www.mezogis.org>)

- Einfache Konfiguration

LAYER

```
CONNECTIONTYPE postgis  
NAME "tablename"  
CONNECTION "user=postgres password=postgres  
dbname=db host=localhost"  
DATA "the_geom FROM table"  
STATUS ON  
TYPE POLYGON  
CLASS  
  COLOR 255 255 200  
END  
END
```


- Syntax entspricht der MapServer-Syntax
- Lediglich 'FILTER' und 'EXPRESSION' sind bei PostGIS-Datenebenen unterschiedlich:
 - Es wird die SQL-Where-Clause anstelle der MapServer-Syntax benutzt
 - FILTER "type = 8"
 - FILTER "name ilike 'pub%' "
 - FILTER "id <= 25 AND height = 20 "
 - EXPRESSION werden klein geschrieben:
 - EXPRESSION ([type] > 2)

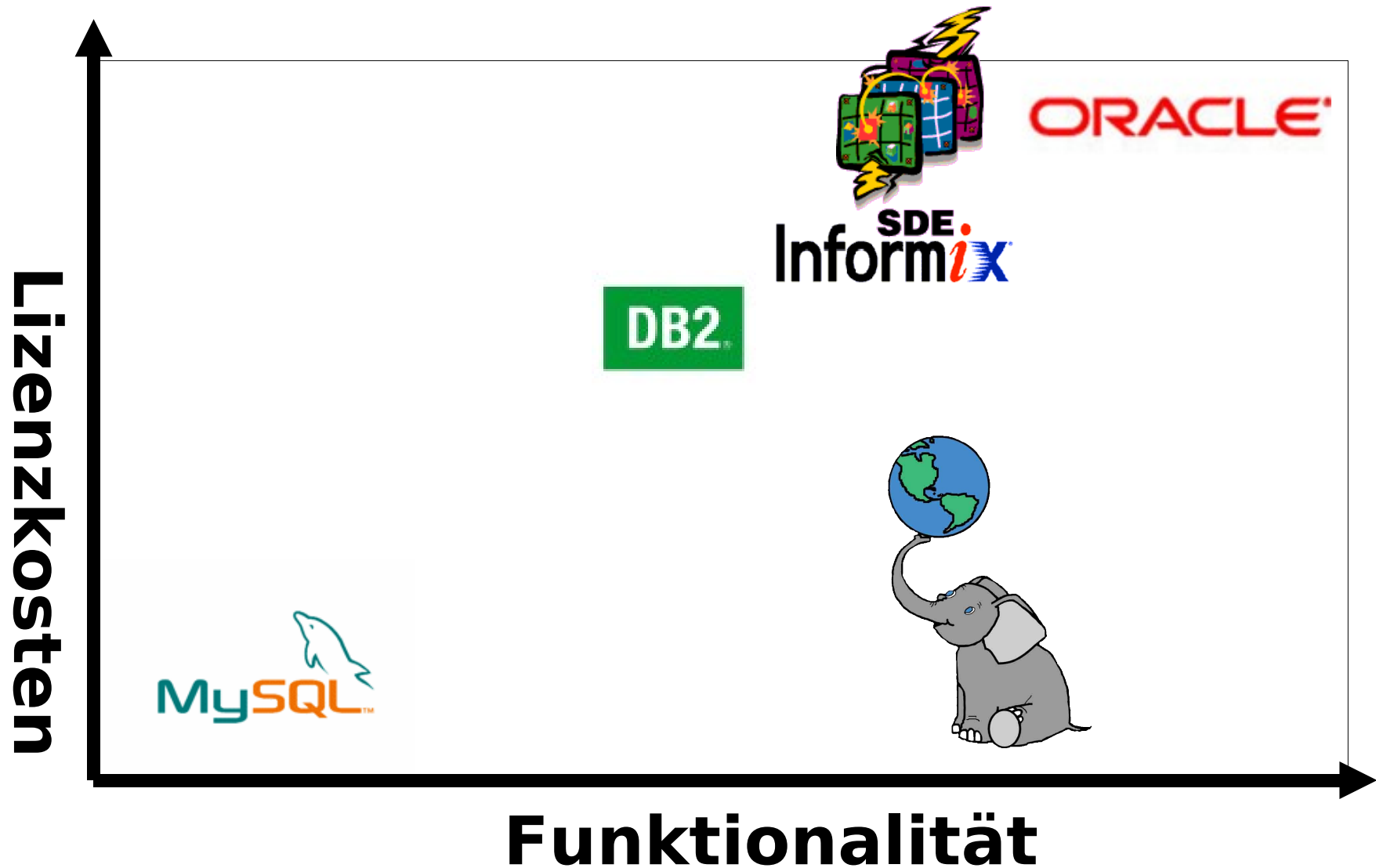
- SQL-basiertes Erzeugen von Geometrien
Zeichnen der 5 längsten Straßen

```
DATA "the_geom from (SELECT name,  
Sum(Length(the_geom)) AS length,  
Collect(GeometryN(the_geom,1))  
AS the_geom FROM table WHERE  
the_geom && setsrid(!BOX!,4326)  
GROUP BY name  
ORDER BY length DESC  
LIMIT 5) as foo  
using SRID=4326, using unique name"
```

- Variablen können durch das CGI-Program von MapServer in der Abfrage ersetzt werden
- Im Template werden die Variablen über hidden-Felder definiert.
 - `<input type="hidden" name="ih" value="[mapheight]">`
- Im Mapfile-Code wird %ih% durch den realen Wert ersetzt

```
DATA "the_geom from
(SELECT the_geom,gid,
distance(the_geom,
geometryfromtext(
'POINT(' ||
(%mx% + %img.x% * %mw% / %iw%)
|| ' ' ||
(%my% - %img.y% * %mh% / %ih%)
|| ')',3005)) AS dist FROM
bc_roads)
as foo using srid=3005 using
unique gid
```

- DATA Anweisung
**DATA "the_geom from (%sql%) as foo
using SRID=4326 using unique gid"**
- Template-Eintrag
**<textarea name="sql" rows=3
cols=50>[sql]</textarea>**
- Nur LAYER TYPE sowie die PROJECTION
muss man im Mapfile vorher definieren.



► Road to PostGIS 2.0 - Wunschliste

- Kreisbodensegmente (Derzeit ist Abspeichern experimentell möglich, es fehlt jedoch an Klienten, die es auch zeichnen können) Limitierungen vorhanden.
- SFSQL2 -> SQL/MM
- Topologie-Unterstützung
- Routing/Networking
- LRS (Linear Reference System)
- Dokumentation (PostGIS Cookbook)

- ▶ www.postgis.org
- ▶ www.postgresql.org

- ▶ Stephan.Holl@intevation.de
- ▶ www.intevation.de/geospatial/

Vielen Dank für Ihre Aufmerksamkeit!

Stephan Holl

<stephan.holl@intevation.de>

Tel.: 0541-3350832