



Správa časoprostorových dat v prostředí PostgreSQL/PostGIS

Antonín ORLÍK

Jan RŮŽIČKA

Josef STROMSKÝ

Pavel DĚRGEL

Jaromír KAMLER





- Co jsou to prostorová data (geodata)
- Kdo a proč je používá
- Vlastnosti geodat
- Formy uložení geodat
- PostgreSQL/PostGIS
- Jak funguje geodatabáze PostGIS
- Import geodat do geodatabáze PostGIS
- Funkce PostGIS
- Struktura DB pro vytvoření a správu dat. skladu
- Metadata
- Čas v geodatech
- Rozhraní nejen pro správu datového skladu
- GIS aplikace s vazbou na PostGIS

Mapy všude kolem nás

- Ubytování pro přednášející na OW 2005:





Mapy všude kolem nás

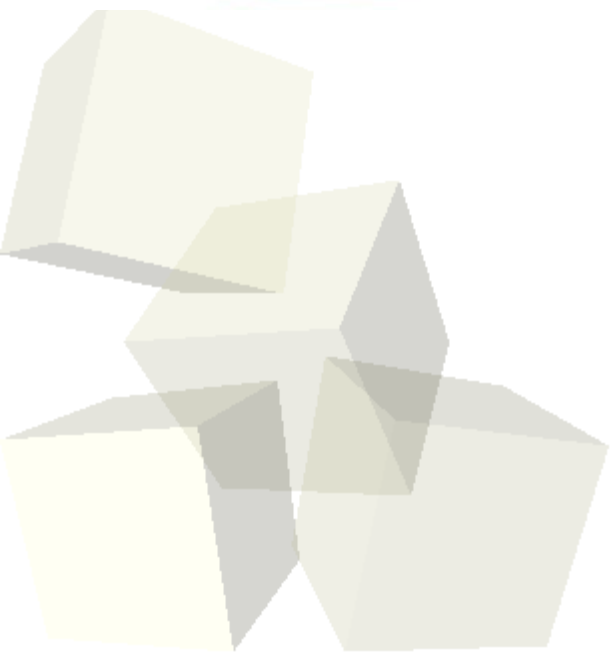
- Ubytování pro přednášející na OW 2005:



WGS-84:
X: 14.38925
Y: 50.10238

S-JTSK:
X: -744851
Y: -1041056

S-42:
X: 5456430
Y: 5552553





Mapy všude kolem nás

■ GPS

MapSource info

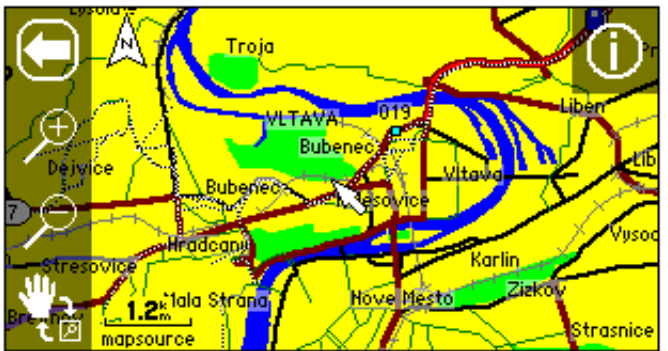
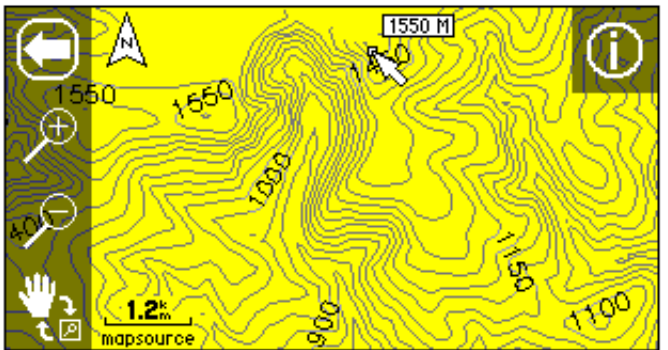
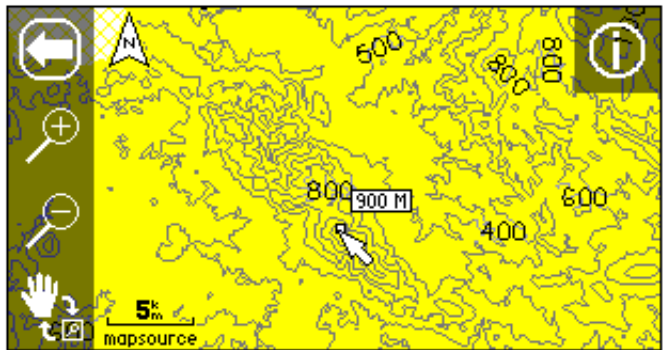
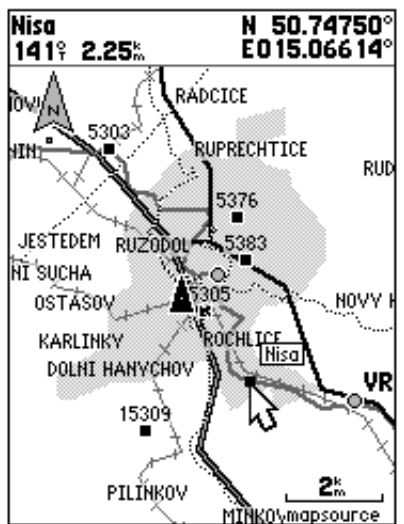
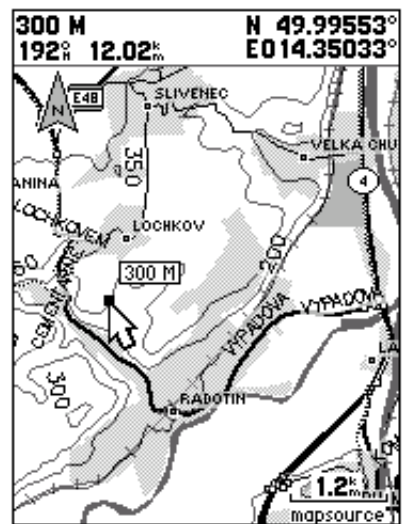
MAP LOADED BY cGPSmapper loader, Led 2004

Ukaz Nazev mapy

TOP050: StC, isolin

Praha, Ceska Repu

Map Set Name: TOP050: StC, isolines 50m, gps-maps.info





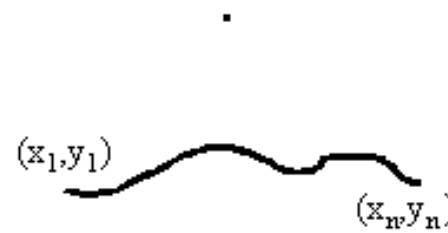
Prostorová data (geodata)

- v oborech lidské činnosti:
 - ◆ doprava, energetika
 - ◆ geologie
 - ◆ vodní hospodářství, meteorologie
 - ◆ vojenství
 - ◆ archeologie
 - ◆ cestovní ruch, turistika
 - ◆ pojišťovnictví
 - ◆ ...a celé řadě zejména přírodních věd
- pro účely:
 - ◆ modelování přírodních a jiných procesů
 - ◆ plánování
 - ◆ vizualizaci
 - ◆ získání nových informací, ...
- různé obory → různé požadavky na data



■ Vektory

- ♦ bod, linie, polygon



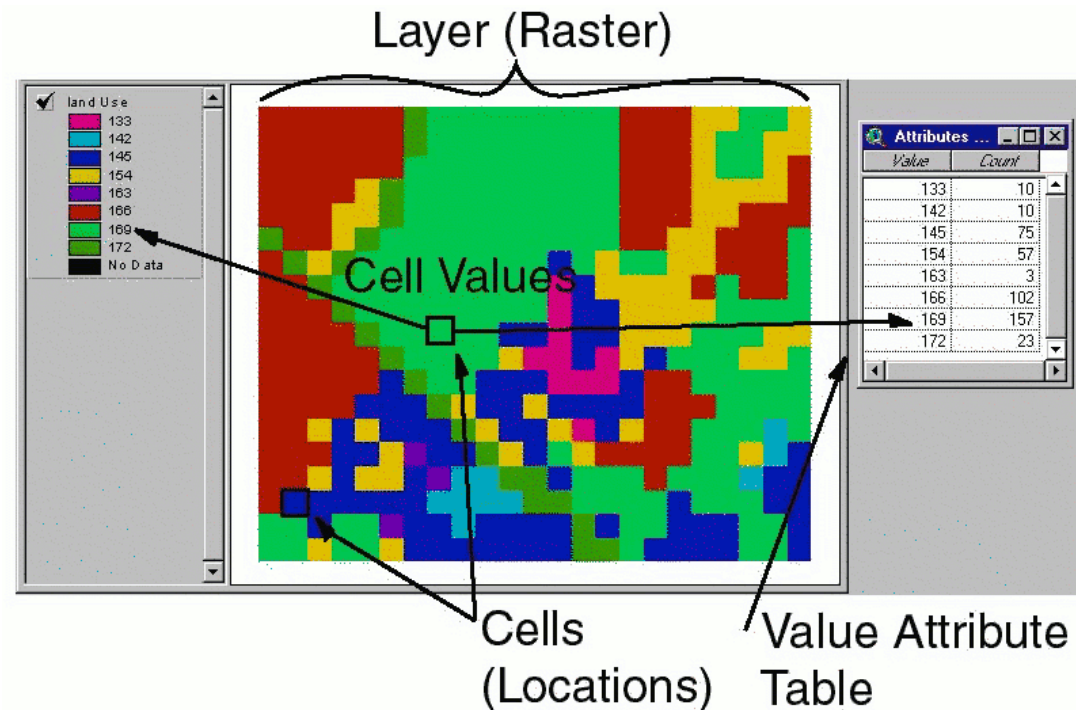
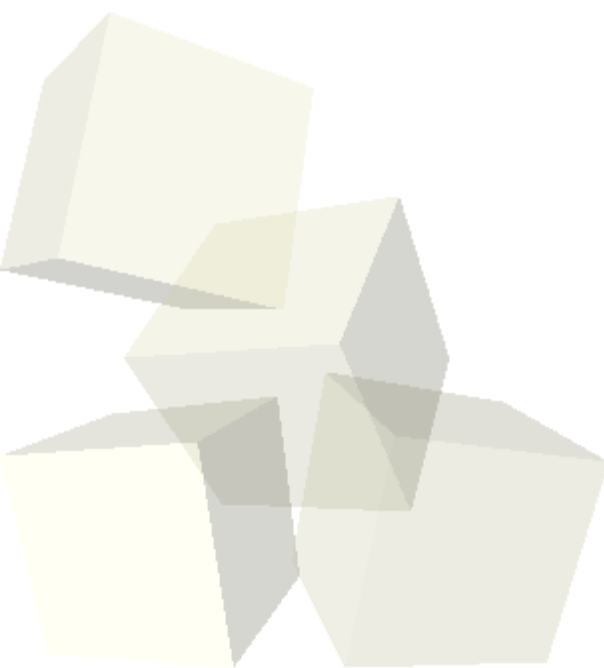
A point (x,y)

A line $(x_1,y_1; x_2,y_2; x_3,y_3; \dots x_n,y_n)$
 x_1,y_1 and x_n,y_n are called nodes; the other points are called vertices



An area $(x_1,y_1; x_2,y_2; x_3,y_3; \dots x_n,y_n;x_1,y_1)$
 x_1,y_1 is called a node; the other points are called vertices

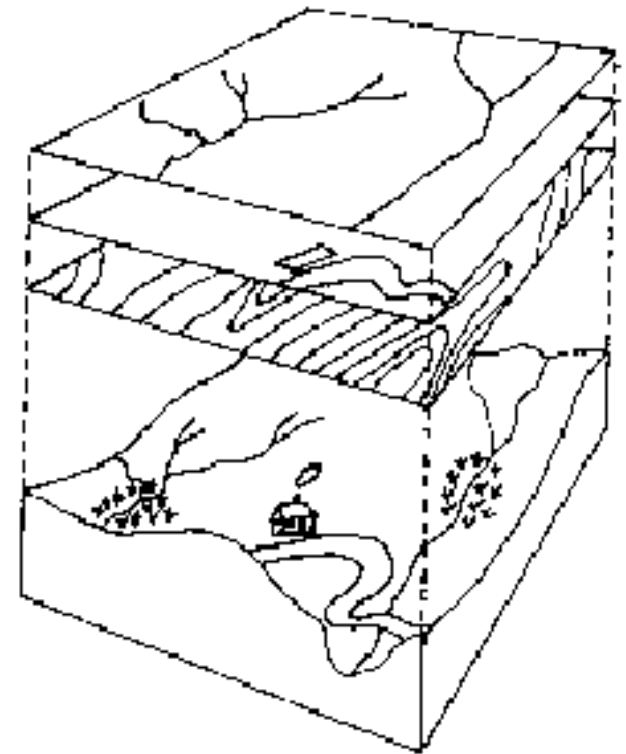
■ Rastry





Vektorová data - vlastnosti

- geografické vymezení (extent)
- souřadnicový systém (referenční systém)
- formát uložení dat
- polohová přesnost
- aktuálnost (časová platnost)
- atributy
- → metadata



Souřadnicový (referenční) systém

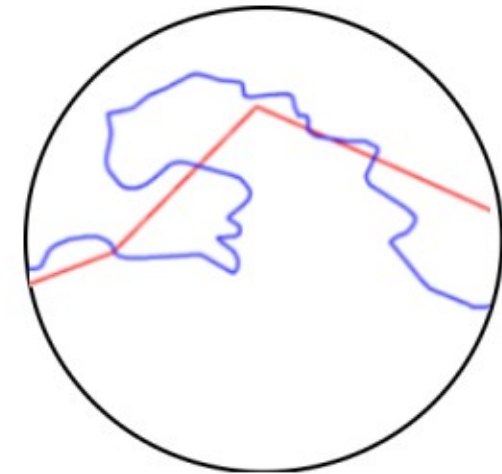
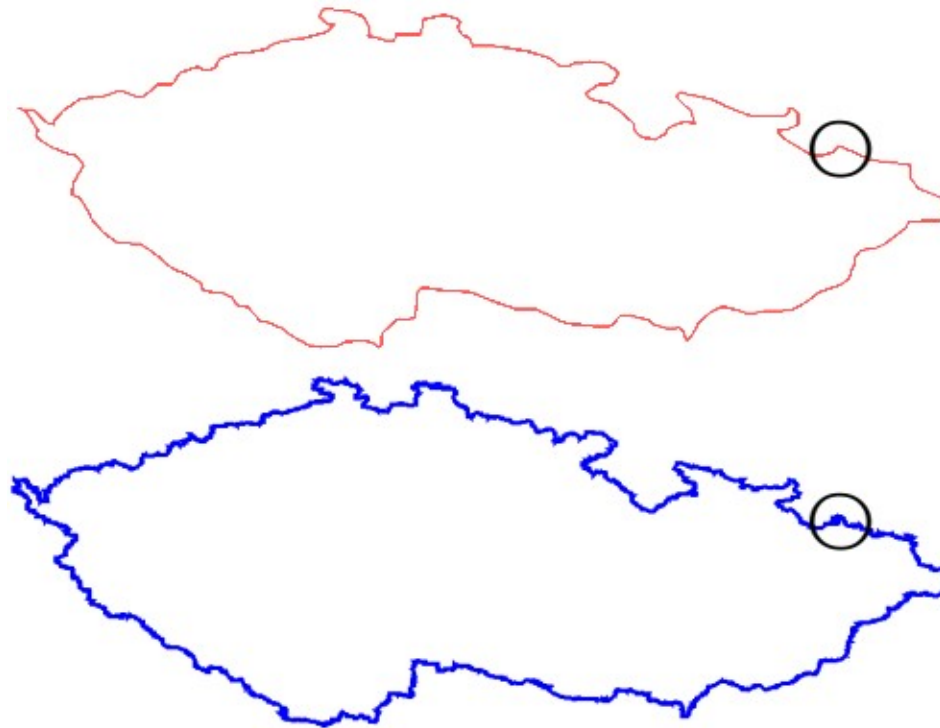
V ČR:

- *S 1942*
- *S-JTSK*
- *WGS-84*
- *WGS-84/NUTM33*



Polohová přesnost

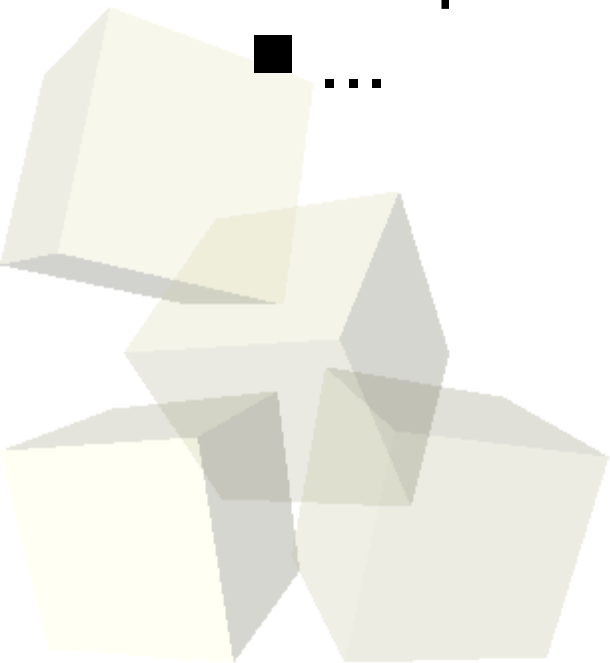
- spojeno s měřítkem použití...





Formáty uložení geodat

- ESRI ShapeFile
 - ◆ *.SHP + *.DBF + *.SHX (+ *.PRJ + *.XML + ...)
 - ◆ vždy jen jedna vrstva
- DWG
- DGN
 - ◆ více vrstev v jednom souboru
- MapInfo
- ...



Souborový způsob ukládání dat





Spása: PostgreSQL/PostGIS

- PostgreSQL – nyní 8.1 beta (8.0.4 stable)
- nadstavba PostGIS:
 - ♦ umožňuje ukládat do databáze i prostorové objekty běžně používané v GIS (2D, 3D, 4D)
 - ♦ rozšiřující funkce → jednoduchá správa a manipulace s těmito objekty
 - ♦ instalace:

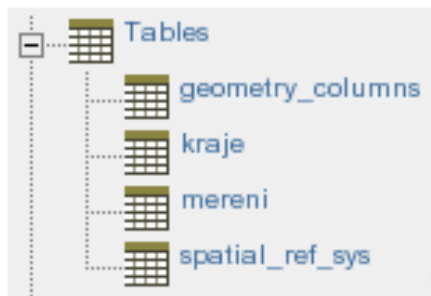
```
cd contrib
gunzip postgis-1.0.0.tar.gz
tar xvf postgis-1.0.0.tar
cd postgis-1.0.0
make
make install
createlang plpgsql yourtestdatabase
psql -d yourtestdatabase -f
lwpostgis.sql
psql -d yourtestdatabase -f
spatial_ref_sys.sql
```




Co se stane?

■ Přibudou:

- ♦ nové datové typy (geometry, ...)
- ♦ nové operátory (&& - průnik geometrií, @ - kompletně obsažen, ...)
- ♦ nové funkce (Distance, Expand, Transform, Extent, ...)
- ♦ nové tabulky:

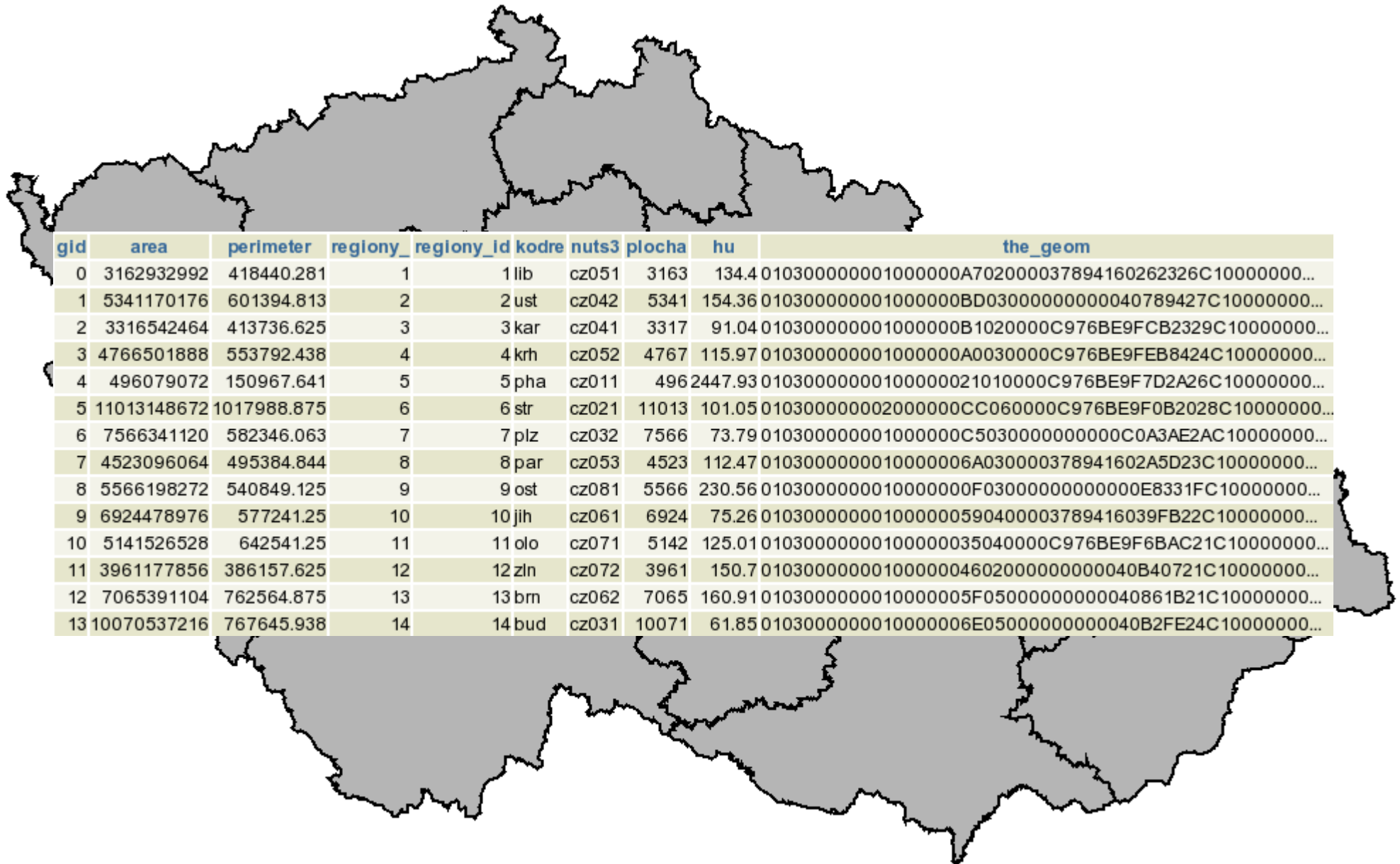


f_table_catalog	f_table_schema	f_table_name	f_geometry_column	coord_dimension	srid	type
	public	kraje	the_geom		2	-1 POLYGON
	public	mereni	the_geom		2 4326	POINT

srid	auth_name	auth_srid	srttext	proj4text
4318	EPSG	4318	GEOGCS["NGN",DATUM["National_Geodetic_Network",SP...	+proj=longlat +ellps=WGS84 +no_defs
4319	EPSG	4319	GEOGCS["KUDAMS",DATUM["Kuwait_Utility",SPHEROID["...	+proj=longlat +ellps=GRS80 +no_defs
4322	EPSG	4322	GEOGCS["WGS 72",DATUM["WGS_1972",SPHEROID["WGS 72...	+proj=longlat +ellps=WGS72 +no_defs
4324	EPSG	4324	GEOGCS["WGS 72BE",DATUM["WGS_1972_Transit_Broadca...	+proj=longlat +ellps=WGS72 +towgs84=0,0,1.9,0,0,0...
4326	EPSG	4326	GEOGCS["WGS 84",DATUM["WGS_1984",SPHEROID["WGS 84...	+proj=longlat +ellps=WGS84 +datum=WGS84 +no_defs
4800	EPSG	4800	GEOGCS["Anguilla_1957",DATUM["Anguilla_1957",SPHE	+semi-longlat +ellps=GRS80 +no_defs



Forma uložení geodat





- z konzole formou SQL
- utilitami - shp2pgsql, ogr2ogr
- pomocí GUI aplikací (QGIS, GRASS, gvSIG, ...)





Import pomocí SQL

```
CREATE TABLE mereni(id int4, name VARCHAR(128));
```

```
SELECT AddGeometryColumn('gdb', 'mereni', 'the_geom', 4326,  
'POINT', 2);
```

```
INSERT INTO mereni (id, name, the_geom) VALUES (1, 'Bod c.  
1', GeometryFromText('POINT(18.053 49.689)', 4326));
```

```
SELECT id, name, the_geom FROM mereni;
```

id	name	the_geom
1	Bod c. 1	0101000020E610000021B07268910D3240D578E92631D84840

```
SELECT id, name, AsText(the_geom) from mereni;
```

id	name	astext
1	Bod c. 1	POINT(18.053 49.689)



Import/export pomocí utilit

Import:

```
shp2pgsql -s 4326 mereni mereni | psql -U username -h  
localhost -d gdb
```

```
ogr2ogr -f PostgreSQL PG:'dbname=gdb user=username'  
mereni.shp
```

Export:

```
pgsql2shp -f mereni -h localhost -u username gdb mereni
```

```
ogr2ogr -f "MapInfo File" mereni PG:'dbname=gdb  
user=username' -sql "SELECT * FROM mereni"
```



Prostorové operátory a funkce

Najdi mi všechny hospody, ze kterých to není do nemocnice dále než 250 m:

```
SELECT h.name, p.name FROM hospitals h, pubs p WHERE  
Distance(h.the_geom, p.the_geom) < 250;
```

Najdi nejjižněji položené město v ČR:

```
SELECT nazev, y(Transform(the_geom, 4326)) AS latitude  
FROM mesta_cr ORDER BY latitude ASC LIMIT 1;
```

Najdi všechna města Moravskoslezského kraje:

```
SELECT nazev FROM mesta_cr WHERE the_geom && (SELECT  
the_geom FROM kraje WHERE kodre = 'ost');
```




Co soubory neumí ;-)

1. Uživatel má vrstvu států světa v souřadnicovém systému WGS-84, formátu ESRI Shapefile a silniční síť Evropy v transverzálním Merkátorově zobrazení, formátu DGN. Rád by z této vrstvy silniční sítě získal pouze silniční síť pro ČR, kterou bude mít v souřadnicovém systému S-JTSK a formátu ESRI Shapefile. Pak uživatel musí:

- převést formát DGN do formátu ESRI Shapefile
- transformovat vrstvu silniční sítě do systému WGS-84
- z vrstvy států světa vybrat ČR a provést překryvnou operaci s vrstvou silniční sítě
- transformovat výslednou vrstvu do souř. systému S-JTSK

v PostGISu:

```
CREATE TABLE silnice_cr AS SELECT Transform(the_geom,  
102065), * FROM silnice_eu WHERE the_geom &&  
Transform((SELECT the_geom FROM staty WHERE  
cntry_name='Czech Republic'), (SELECT srid FROM  
geometry_columns WHERE f_table_name='silnice_eu'))
```



Řízení přístupu uživatelů

■ Vytvoření DB uživatele:

- `CREATE USER user_xy WITH PASSWORD 'password';`

■ Právo čtení:

- `GRANT SELECT ON TABLE kraje TO user_xy;`
- `GRANT SELECT ON TABLE geometry_columns TO user_xy;`

■ Aktualizace dat:

- privilegium UPDATE, INSERT a DELETE k příslušné tabulce obsahující prostorová data

■ Zápis dat:

- `GRANT CREATE ON DATABASE gdb TO user_xy;`
- `GRANT INSERT ON TABLE geometry_columns TO user_xy;`

■ Skupina uživatelů a výběr jejich členů:

- `CREATE GROUP skupina_x WITH USER user_xy, user_z;`
- `ALTER GROUP skupina_x ADD USER user_y;`

■ Právo pro skupinu:

- `GRANT SELECT ON TABLE kraje TO GROUP skupina_x;`



Hlavní MENU

- [Spolupráce s EU](#)
- [Novinky](#)
- [Přihlásit se](#)
- [Zaregistrovat se](#)
- [Dokumentace](#)
- [Nápověda](#)
- [O autorech](#)
- [Vaše postřehy](#)
- [MIDASLite](#)
- [MIDAS do praxe](#)
- [Další nástroje](#)
-
- [Úvodní stránka](#)
- [Win 1250](#) | [8859-2](#)



Podmínky získání a/nebo užití, možnosti využití

? Podmínky užití/využití - kategorie	se souhlasem nebo na základě smlouvy, licenční poplatky, nesmí být prodáváno bez povolení, dodávání je licensováno, vývoz do zahraničí může vyžadovat zvláštní povolení
? Podmínky užití/využití	
? Podmínky užití/využití (www)	
? Způsob a podmínky získání	
? Způsob a podmínky získání (www)	
? Distribuční jednotky	město
? Velikost distribuční jednotky	
? Cena	dle aktuálního ceníku http://www.ceda.cz/cenik.htm
? Nosiče pro přenos dat	CD-ROM, počítačová síť
? Formáty	BMP (MS-Windows bitmap), DBF (III, IV, V), MID (MapInfo), MIF (MapInfo), ArcView Shape soubor, Text s oddělovači
? Poznámka k formátům	Po konzultaci lze vytvořit i některé další formáty

Údaje o metadatatech

? Datum vzniku metadat	15.07.2003
? Datum poslední aktualizace metadat	05.04.2004
? Datum očekávané aktualizace metadat	

Organizace

Central European Data Agency, a.s.	(Původce/Autor, Vlastník, Administrator, Poskytovatel, Prodejce, Správce metadat, Vlastník autorských práv, Kontaktní místo, garant/gestor)
--	---

Osoby

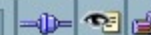
[Vaněk](#) (Kontaktní osoba)

Datové soubory

Není žádný záznam

Objekty datových souborů

Není žádný záznam





■ Nejčastější dotazy:

- ♦ název
- ♦ textový popis
- ♦ původ
- ♦ přesnost
- ♦ aktuálnost (platnost dat)
- ♦ souřadnicový systém
- ♦ extent
- ♦ datový typ (bod, linie, polygon, rastr)
- ♦ vlastník

■ → tabulka DATA:

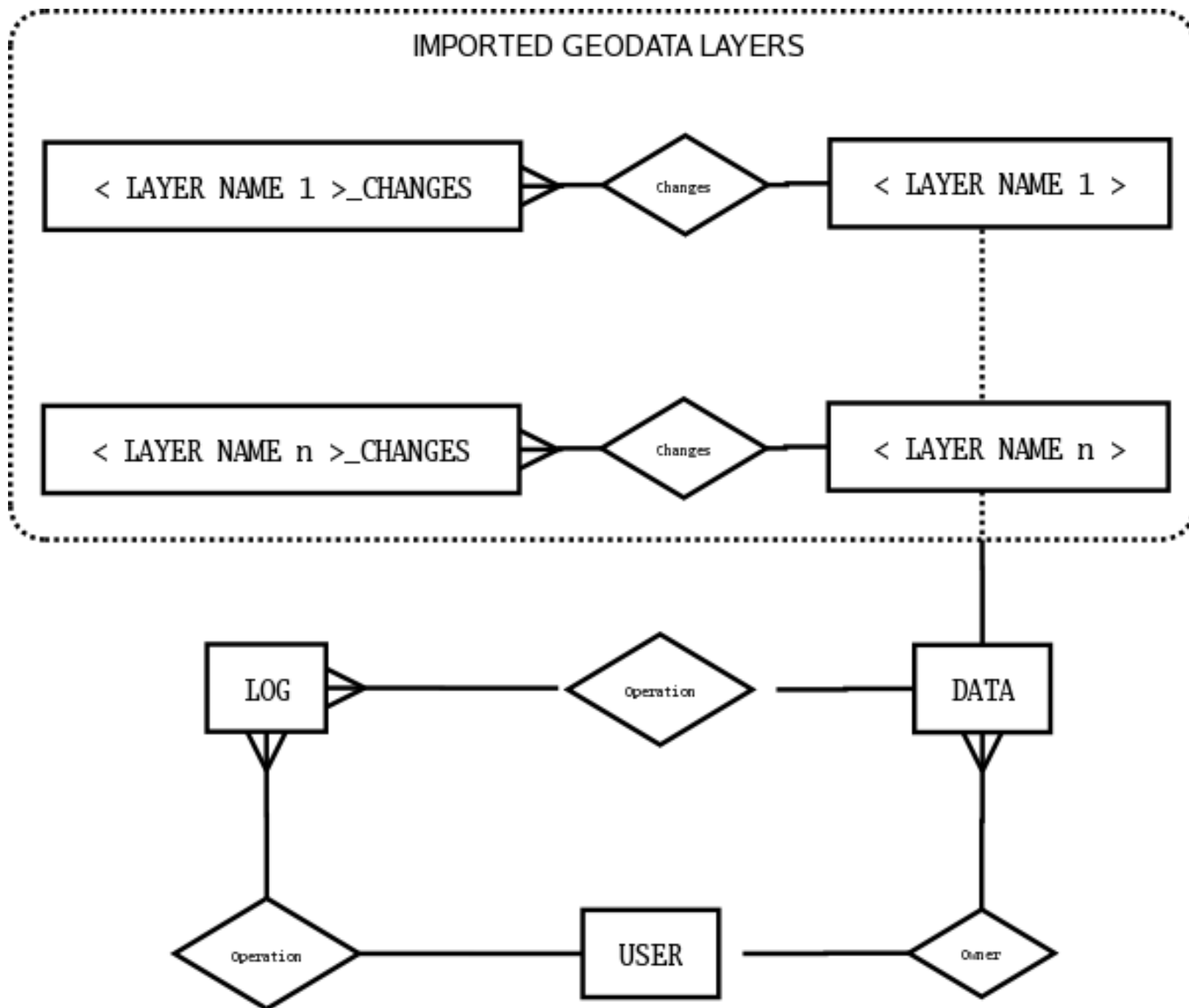
- ♦ základní informace o datech (↑) + odkaz na úplný metadatový záznam v systému MIDAS



- Aktualizace:
 - ♦ např.: název ulice, majitel pozemku, rozdělení parcel, pohyb vozidla, ...
 - ♦ → UPDATE, INSERT, DELETE
- Vhodné uchovávat původní data (důvodů mnoho)
 - ♦ platnost od
 - ♦ platnost do
 - ♦ čas a datum aktualizace
 - ♦ kdo provedl aktualizaci
- Neuchovávat celou datovou sadu, ale jen ten geoprvek nebo atribut, kde došlo ke změně
- Aktuálně uchovávat poslední platnou vrstvu a mít možnost vygenerovat vrstvu k datu dřívějšímu



Struktura datového skladu





- PostGIS umí i 4D – ale omezeně:
 - ♦ POINT(15.8 19.5 385 15)
- Použít 3D jako 2D + čas?
 - ♦ který čas zapsat?
 - čas aktualizace || čas konce platnosti původních dat?
 - co když chci evidovat obojí?
 - kam zapíšu uživatele, který provedl aktualizaci?
- Jak evidovat změny?
 - ♦ určitě: GID, operace (INSERT, DELETE, UPDATE), uživatel, čas aktualizace, platnost od – do
 - ♦ u UPDATE zaznamenat jen změněný atribut nebo celý geoprvek? (i geometrie je zde atributem)
 - ♦ použít zvlášť tabulku pro UPDATE, kde budou jen změněné atributy a tabulku pro INSERT + DELETE, kde bude celý geoprvek?
 - ♦ Jak náročná bude rekonstrukce?



Jak nyní hledat data?

Souborový způsob:

Uživatel má na svém pevném disku svého počítače velké množství nejrůznějších geodat (třeba i přehledně roztríděných ve vhodně pojmenovaných adresářích). Nyní se stane jistá oblast, třeba okolí obce Mrákotín, středem jeho zájmu a rád by zjistil, jaká data má v této oblasti k dispozici.

→ uživatel musí pomocí vhodné prohlížečky geografických dat jednotlivé vrstvy načíst, zjistit, zda daná vrstva leží v zájmové oblasti, zda má dostatečnou polohovou přesnost, zda je aktuální a dále musí zjistit, zda ještě nemá na jiném místě uloženy nějaké vrstvy v jiných souřadnicových systémech.

PostGIS:

- např. vytvořením univerzálního skriptu, který vrátí popis vrstev v uživatelem definované oblasti (díky prostorovému dotazu nad množinou všech dostupných vrstev) + možnost atributových omezení... (viz. dále)



Hledání dat v zadané oblasti

```
#!/usr/bin/perl
use DBI;

# Arguments: boundary coordinates, epsg code of coordinate system
($west, $south, $east, $north, $user_epsg) = @ARGV;

my $dbh = DBI->connect('dbi:Pg:dbname=gdb;host=localhost', 'user', 'password')
    or die $DBI::errstr;

# Select all imported layers and their epsg code
$sth1 = $dbh->prepare( "SELECT layer, epsg, description, lineage, accuracy, \
    validf, validt, datatype FROM data" );
$sth1->execute();
print "These layers overlaps inserted rectangle:\n\n";
while ( @row1 = $sth1->fetchrow_array ) {
    $layer = $row1[0];
    $db_epsg = $row1[1];
    # Create rectangle from inserted boundary coordinates, transform it to coordinate
    # system of current layer and select all records that overlaps the rectangle.
    $sth2 = $dbh->prepare( "SELECT gid FROM $layer WHERE the_geom && \
        Transform(GeomFromText('POLYGON ($west $south, $east $south, \
        $east $north, $west $north, $west $south)', $user_epsg), $db_epsg)" );
    $sth2->execute();
    @row2 = $sth2->fetchrow_array;
    # Empty result mean that there are no data in inserted rectangle.
    if ( $row2[0] ne "" ) {
        print "Layer: $layer\n";
        print "\tDescription: $row1[2]\n";
        print "\tLineage: $row1[3]\n";
        print "\tAccuracy: $row1[4] meters\n";
        print "\tEPSG code: $db_epsg\n";
        print "\tValid from: $row1[5]\n";
        print "\tValid to: $row1[6]\n";
        print "\tData type: $row1[7]\n\n";
    }
}
__END__
```

GUI pro datový sklad

User: root [;] Language: (CZ) - Mozilla

System Tools Compositions Help

Jednotlivé budovy

Bloky budov a průmyslové areály

Řeky, vodní nádrže

močál/bažina, vodní nádrž, mokřina

Řeka/potok, kanál, pramen

Vrstevnice

Plošné porosty a sam. stojící stromy

Strom/Skupina stromů

Les/Lesní školka

Pastvina, louka, sad, chmelnice, rákos

System successfully started

0 0.4 0.8 1.2 1.6 km

Transferring data from transcat.vsb.cz...

vojenský geografický a hydrometeorologický úřad

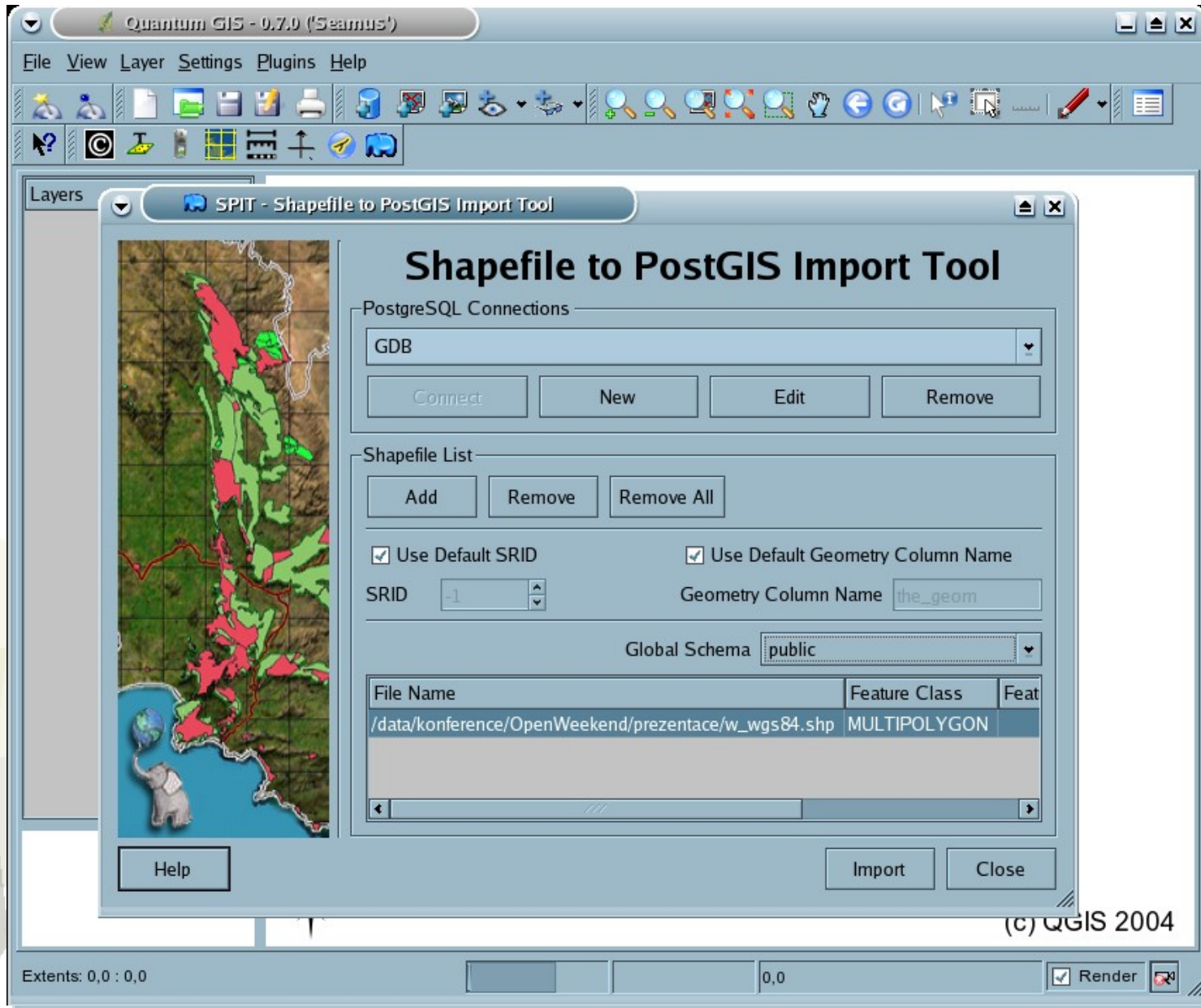


GUI plug-ins aneb zásuvné moduly ;-)

- registrace uživatelů
- import dat do PostGISu + zadání metadat
- přidělování práv k datům
- vytváření skupin
- vyhledávání na základě metadat a prostor. dotazů
- export dat (výřez, transformace do zvoleného souř. systému, volba výstupního formátu)
- zobrazení dat v mapovém okně
- žádost o přidělení práv k datům
- zaslání zapomenutého hesla
- kontrola tzv. mrtvých dat
- kontrola úplnosti metadat (po importu z jiného prostředí)

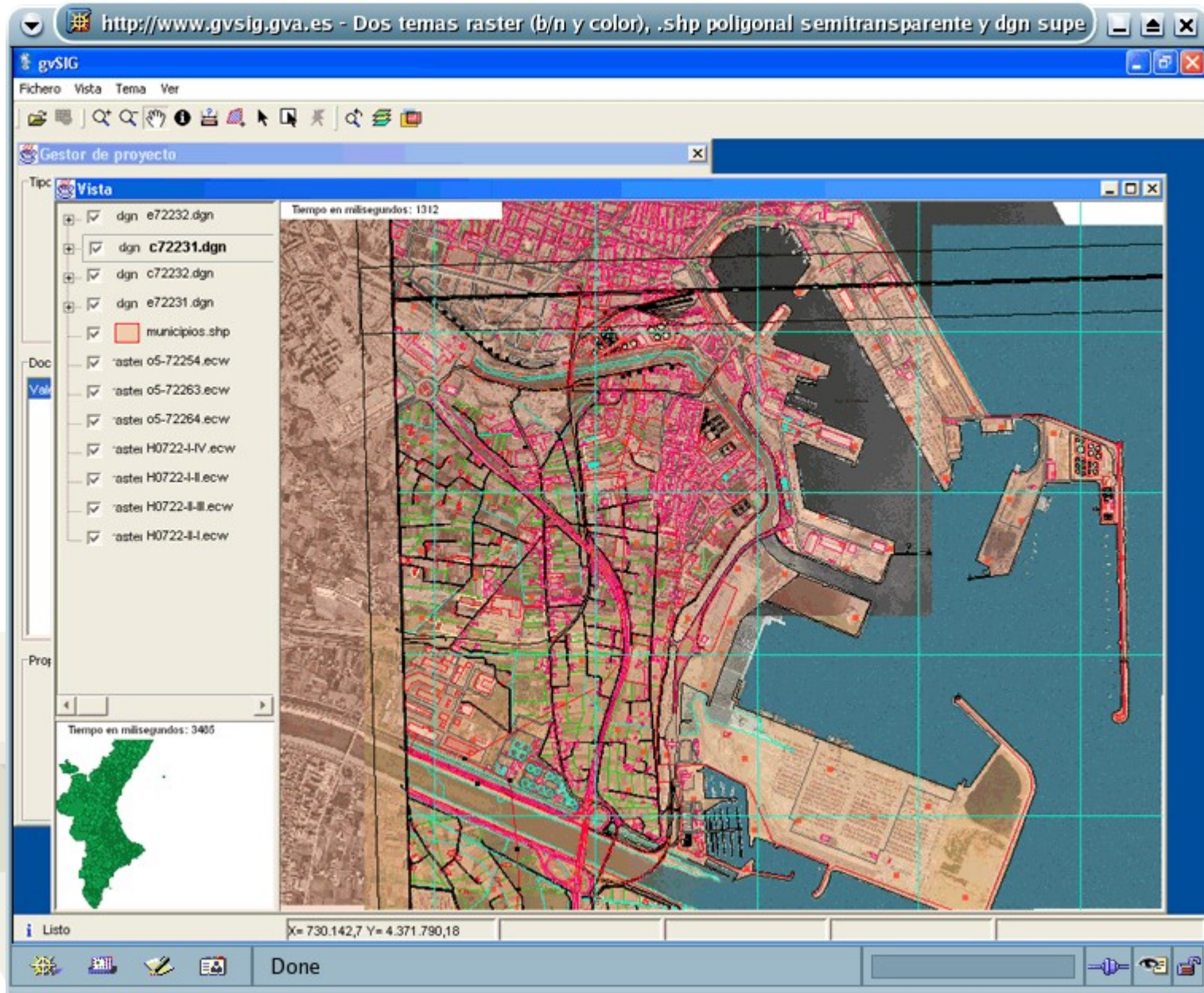
Jiné aplikace podporující PostGIS

■ QGIS (Quantum GIS)



Jiné aplikace podporující PostGIS

■ gvSIG



■ GRASS GIS

The screenshot displays the GRASS GIS 6.1 interface. The main window shows a map of a mountainous region with a red boundary. The interface includes a menu bar (File, Config, Raster, Vector, Image, Grid3D, Databases, Help) and a toolbar with various icons. The Vector Manager panel on the right shows the following settings:

- Vector name: `polbnda_italy@PERMANENT`
- Display: shapes categories topology line directions
- points lines boundaries centroids areas faces
- Point symbols: icon `basic/x` size `5`
- Draw lines: color `red` width `1` (pixels)
- Fill areas: color random colors GRASSRGB column colors
- Label vectors: label color `black` size `8` align with pt `left` center `center`
- layer for labels `1` attribute col for labels `vmap0_nam`
- Query vectors: layer for query `1` query cat values SQL query
- SQL where statement
- Mouse query setup: edit attributes (form mode) results as text in terminal
- Display when avg. region dimension is `>` or `<`
- Line width for ps.map print output: `1`



■ Nejjednodušší webový mapový klient:

The screenshot displays a web map interface for a map of Europe. On the left side, there is a control panel with the following elements:

- Zoom In: A magnifying glass icon with a plus sign.
- Zoom Out: A magnifying glass icon with a minus sign.
- Pan: A hand icon.
- Full Extend: A button with the text "Full Extend".

Below the control panel is a dashed line and a legend section:

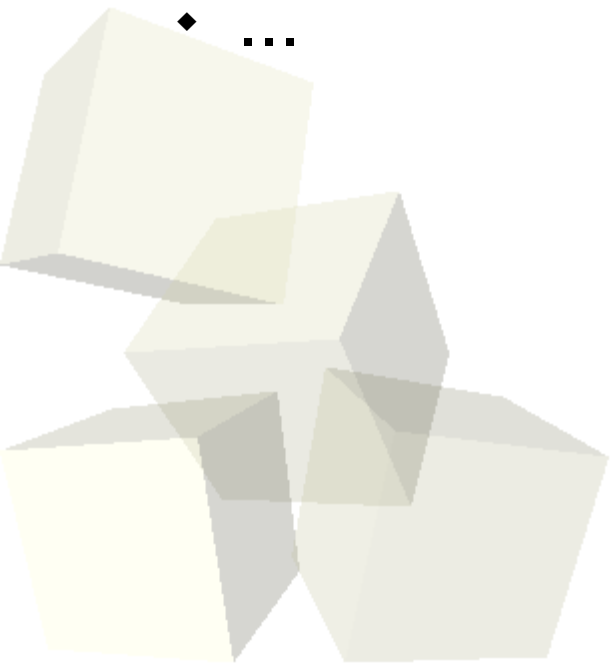
Legenda:

- Capital City: Represented by a red dot.
- Country: Represented by a black outline.
- Raster background: Represented by a green and brown textured area.

The map itself shows a topographic view of Europe with various countries and their capital cities labeled. Countries shown include Denmark, Lithuania, Belarus, Poland, Ukraine, Moldova, Romania, Bulgaria, Hungary, Slovakia, Austria, Slovenia, Croatia, Yugoslavia, Italy, Switzerland, France, Germany, Netherlands, Belgium, Luxembourg, Czech Republic, and Monaco. Capital cities marked with red dots include Berlin, Warszawa, Vilnius, Minsk, Praha, Wien, Bratislava, Budapest, Bern, Vaduz, Ljubljana, Zagreb, Beograd, Sarajevo, and Bucuresti. A scale bar at the bottom indicates distances from 0 to 1000 km.



- Výhody datového skladu PostGIS
 - ◆ data uložena centrálně
 - ◆ data uložena na jednom místě zobrazována na mnoha jiných místech (+ web)
 - ◆ zachování práv
 - ◆ nezávislost na klientském prostředí
 - ◆ systém žije sám (minimální administrátorské zásahy)
 - ◆ dodržení standardů pro ukládání geodat
 - ◆ ...





Děkuji za pozornost...

(VŠB - Institut geoinformatiky: <http://gis.vsb.cz>)
(Den GIS: http://gis.vsb.cz/Den_GIS/)

