

Geografické informační systémy

Databázové systémy pro GIS

Databázové systémy pro GIS

- ◆ Struktura a funkce databázových systémů/uložení dat v GIS.
- ◆ Různé varianty uložení.
 - ◆ Především: *Atributové údaje*.
 - ◆ Dnes stále více i zkoumání uložení geometrie.
 - ◆ Uložení topologie (stále vnímáno jako cosi „navíc“).
- ◆ V předchozí přednášce se probíraly datové modely uložení vektorových a rastrových dat.
- ◆ Jedno z pojetí GIS (až příliš technokratické):
“GIS je databáze”

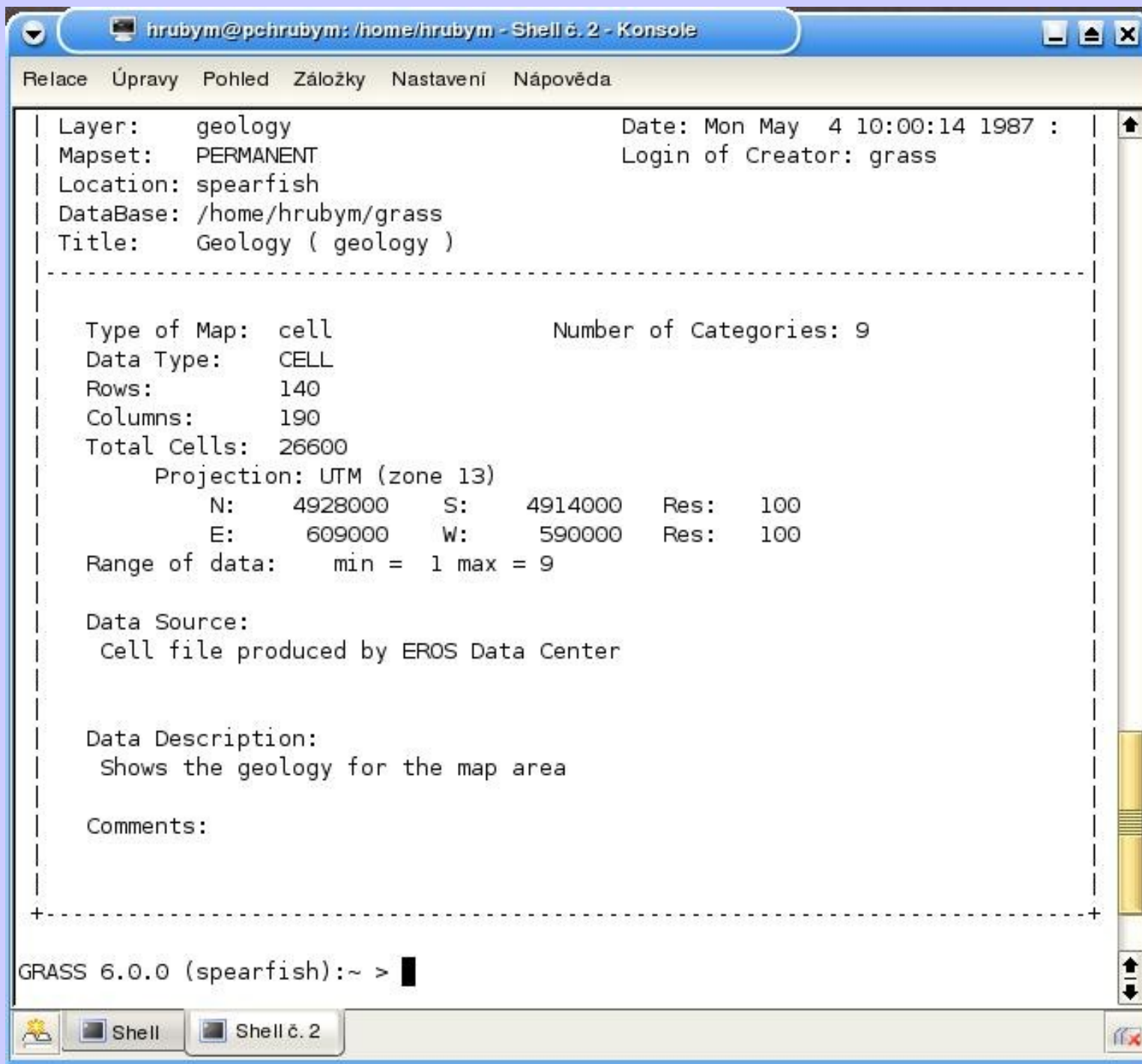
Základní dělení

- ◆ souborové systémy – tradiční přístup
- ◆ soubory + databáze – současný přístup
- ◆ databáze – k tomu se blížíme....(včetně *rastrů*)
- ◆ databáze – směřují na servery
- ◆ metadata – informace o podobě dat (data o datech)
- ◆ Problémy:
 - ◆ správa vrstev – množství souborů
 - ◆ sdílení vrstev – GIS servery (souborové, služby)

Co potřebujeme ukládat

- ◆ Vektor
 - ◆ metadata - komentáře
 - ◆ prostorová informace – geometrie, topologie
 - ◆ atributy
- ◆ Rastr
 - ◆ metadata – rozměr prostoru, způsob dělení prostoru, reklasifikace, ...
 - ◆ posloupnost buněk (atributů) – fakticky řada čísel, dlaždice
- ◆ Metadata -- <http://gis.vsb.cz/midas/>
- ◆ Metadata jsou v současných GISech velmi důležité téma – INSPIRE
- ◆ Zveřejňování metadat – stále nevyřešený problém

Metadata v rastru



```
hrubym@pchrubym: /home/hrubym - Shell č. 2 - Konsole
Relace  Úpravy  Pohled  Záložky  Nastavení  Nápověda

Layer:      geology          Date: Mon May  4 10:00:14 1987 :
Mapset:     PERMANENT        Login of Creator: grass
Location:   spearfish
DataBase:   /home/hrubym/grass
Title:      Geology ( geology )

-----

Type of Map: cell          Number of Categories: 9
Data Type:  CELL
Rows:       140
Columns:    190
Total Cells: 26600
Projection: UTM (zone 13)
            N:  4928000  S:  4914000  Res:  100
            E:  609000  W:  590000  Res:  100
Range of data:  min = 1 max = 9

Data Source:
Cell file produced by EROS Data Center

Data Description:
Shows the geology for the map area

Comments:

GRASS 6.0.0 (spearfish):~ > █
```

Souborové systémy

- ◆ Uložení prostorových a atributových dat (pokud jsou) v souborech.
- ◆ GIS obsahuje sadu funkcí pro manipulaci s daty v souborech (podpora formátů).
- ◆ Problém při víceuživatelském přístupu.
Nebezpečí poškození dat (integrita, nezávislost, bezpečnost)
 - ◆ GRASS: soubory a lokace.
 - ◆ GRASS – ryze souborový systém (je abstrahováno uživateli)
 - ◆ ArcGIS – značně souborový, ArcCatalog, Geo-databáze

Databázové modely

- ◆ Relační - operace nad relacemi, klíče
 - ◆ (známé z předchozích předmětů)
- ◆ Postrelační – lze definovat složitější datové typy.
 - ◆ Lze mít záznamy různé délky (multi-linie).
- ◆ Objektově orientované modely – GIS nástroj SmallWorld

geoSQL – prostorové operátory

- ◆ select where
- ◆ distance(o1,o2), buffer(obj, dist)
- ◆ o1 inside o2
- ◆ o1 contain o2
- ◆ o1 cross o2
- ◆ ...
- ◆

První generace GIS - Systémy bez atributových souborů.

- ◆ Čistě rastrový přístup neodděluje polohové a atributové informace. Rastr=jeden soubor.
- ◆ Nepřítomnost DB systému v rastrovém přístupu není omezující, naopak - uložení je jednodušší...
- ◆ Problém nastává při kombinování více rastrových vrstev - mnoho souborů (počet souborů je dán počtem potřebných atributů)

První generace GIS - Flat soubory

- ◆ Flat soubory - jednoduché tabulky údajů. Všechny mají stejnou strukturu.
 - ◆ Individuální tabulky v relačním modelu se mohou nazývat flat soubory.
- ◆ prostorové údaje se nemusí opakovat
 - ◆ náznak druhé generace
 - ◆ soubor s polohami + sada souborů s atributy, ...
- ◆ GIS nástroj IDRISI

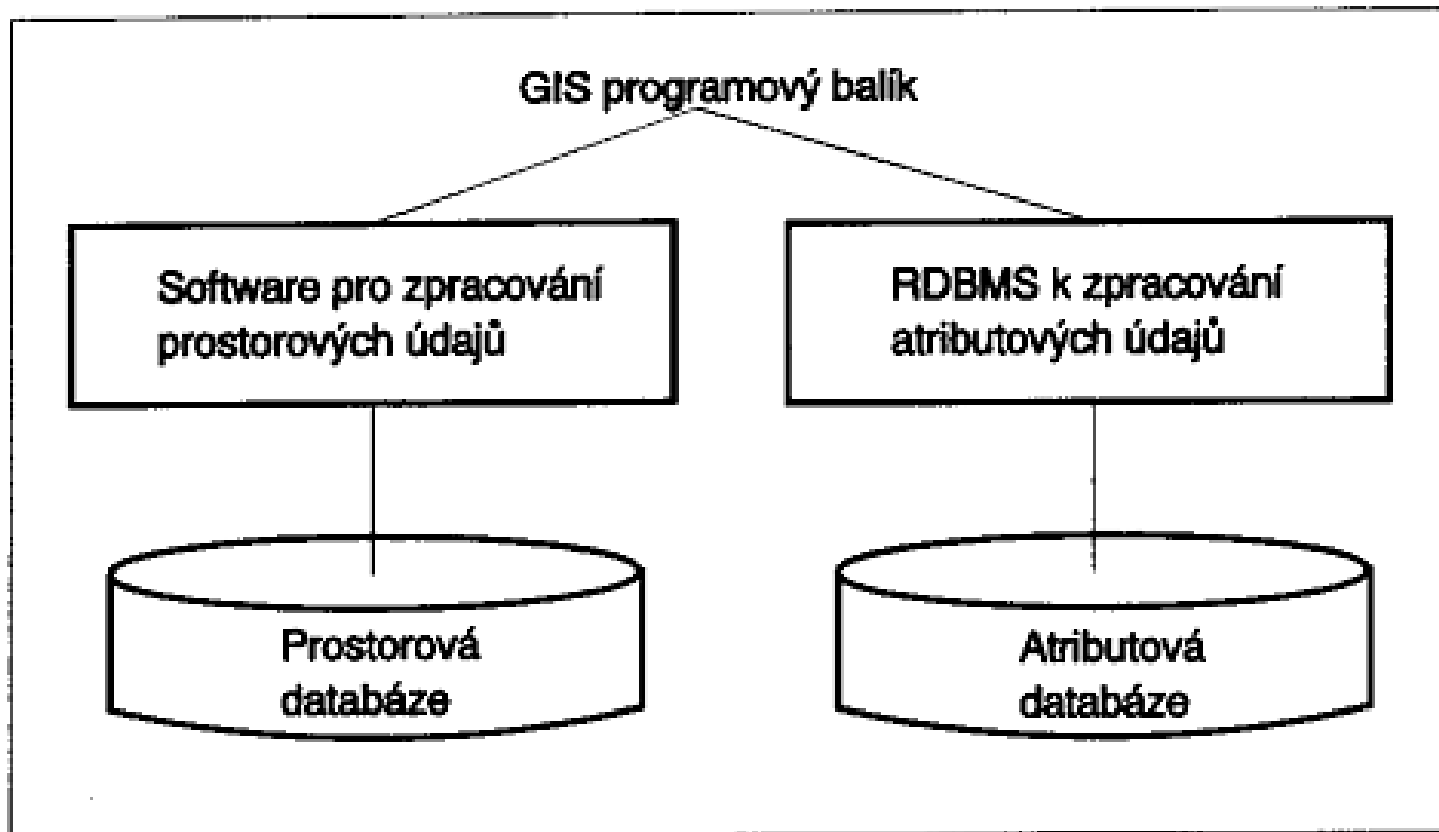
Druhá generace GIS, duální systémy

- ◆ Relaçní přístup. V podstatě nejrozšířenější přístup (Arc/Info, Systém 9, Geo/SQL).
- ◆ V tabulkách se ukládají:
 - ◆ pouze atributové údaje,
 - ◆ veškerá data
- ◆ Vektorový objekt má unikátní ID (cat)
 - ◆ „cat“ je klíč do napojené tabulky
- ◆ **Celkově vzato je tento přístup kompromisní a netechnický**

- ◆ Pak se nabízí dvojí použití spojení polohová/atributová data:
 - ◆ k objektům dodávají popisky (jména řek...)
 - ◆ systém zobrazuje objekty na základě DB dotazu (zobraz města s počtem obyvatel...)
- ◆ Více v přednášce o analýzách:
 - ◆ prostorové
 - ◆ atributové dotazy

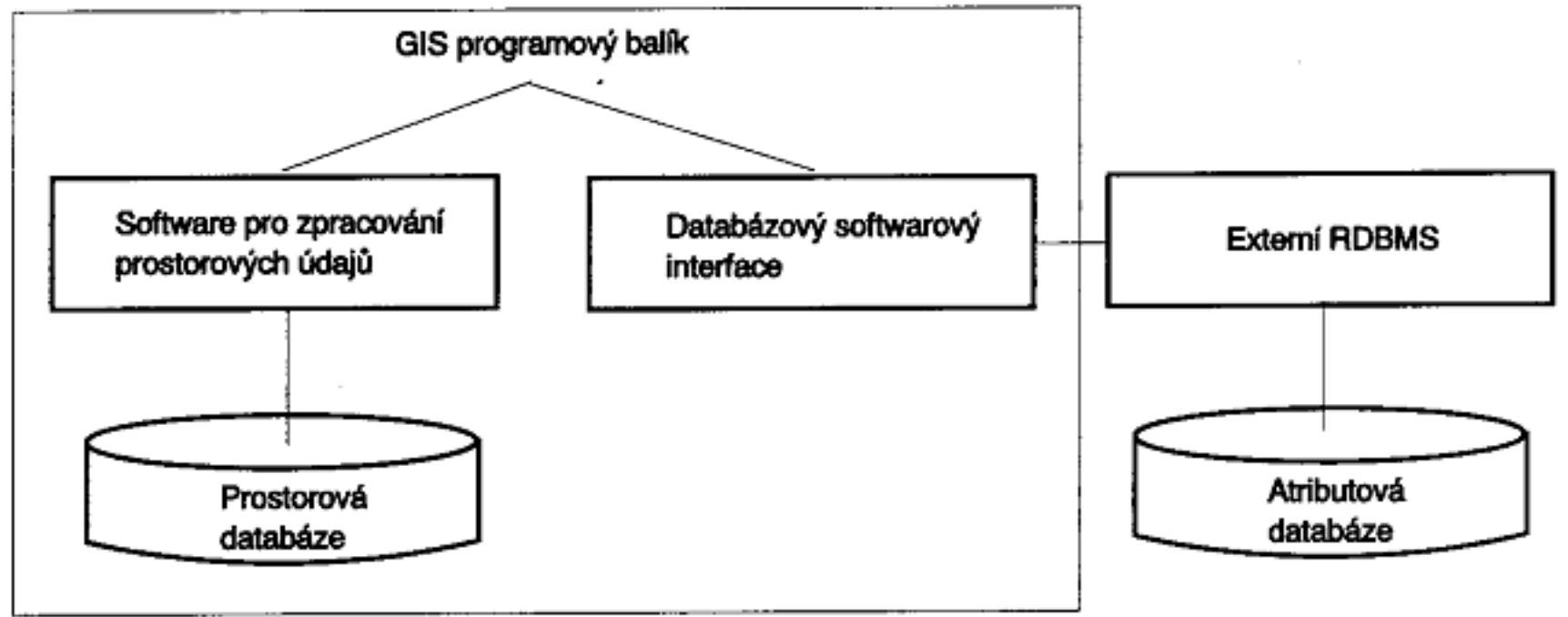
Duální s., DBMS implementovaný do GIS

a) DBMS implementovaný do GIS



Duální systémy, externí DBMS

b) Externí DBMS



III. generace GIS (postrelační)

- ◆ Relacionální datový model je poměrně omezený.
- ◆ Novinky: (ideální požadavky na rozšíření)
 - ◆ možnost používat záznamy různé délky
 - ◆ možnost definovat komplexní objekty
 - ◆ možnost vyměňovat kompletní GIS databáze mezi různými GIS systémy
 - ◆ podporuje rastrovou i vektorovou reprezentaci
 - ◆ *prostorové operátory jako například vzdálenost, plocha, obvod*
 - ◆ *možnost prostorových indexových souborů*
- ◆ **Bez tak je atributový údaj nějak kódovaný BLOB.**
 - ◆ Důležitější je koncepce a abstrakce vrstvy.

PostgreSQL + PostGIS

- ◆ (použity materiály Antonína Orlíka, VŠB)
- ◆ Postrelační DB systém se “spatial” nádstavbou
 - ◆ datové typy (bod, linie, polygon)
 - ◆ implementace referenčních elipsoidů, souřadných systémů
 - ◆ analytické funkce (vzdálenost, průniky, ...)
- ◆ <http://www.postgresql.org/>

PostGIS

- ◆ <http://www.postgis.org/>
- ◆ Napojené SW: MapServer, GeoTools, GeoServer, GRASS, JUMP, QGIS, OGR
- ◆ geometrie – bod, linie, polygon, ...
- ◆ ?ukládání topologie?
 - ◆ další .sql rozšíření
 - ◆ V zásadě špagetový model typu Shapefile.

Instalace/použití

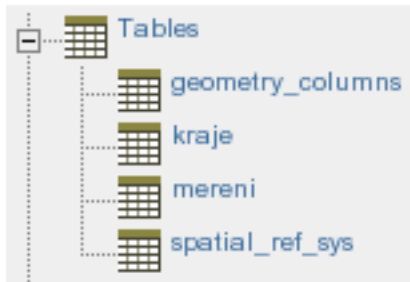
- ◆ Instalujte Postgres
- ◆ Instalujte PostGIS
 - ◆ instalace spočívá v provedení konfiguračních SQL skriptů
 - ◆ skripty zavedou tabulky
 - ◆ tabulka souřadných systémů (spatial_ref_sys)
 - ◆ tabulka tabulek s prostorovým rozšířením (geometry_columns)

Instalace PostGISu

- ◆ createdb mojeDB
- ◆ createlang -U postgres plpgsql mojeDB
- ◆ cd /usr/local/pgsql/lib/ (typicky)
- ◆ psql -U postgres -d mojeDB -f lwpostgis.sql
- ◆ psql -U postgres -d mojeDB -f spatial_ref_sys.sql
- ◆ psql -U postgres -d mojeDB -f esri_extra.sql
- ◆

- ◆ `psql -c 'GRANT ALL ON TABLE geometry_columns TO login' -U postgres -d mojeDB`
- ◆ `psql -c 'GRANT ALL ON TABLE spatial_ref_sys TO login' -U postgres -d mojeDB`

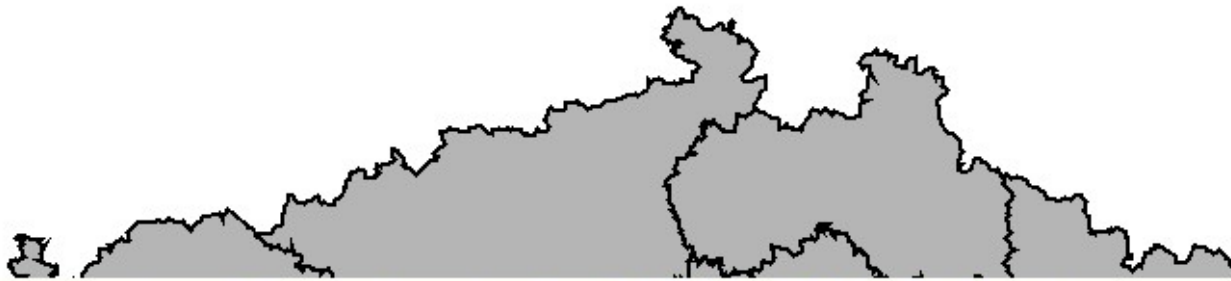
Zavedení PostGISu



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
4600	EPSG	4600	GEOGCS["Aquillo_1957",DATUM["Aquillo_1957",SPHE...	+proj=longlat +ellps=GRS80 +no_defs

Příklad



gid	area	perimeter	regiony_	regiony_id	kode	nuts3	plocha	hu	the_geom
0	3162932992	418440.281	1	1	lib	cz051	3163	134.4	010300000001000000A702000037894160262326C1000000...
1	5341170176	601394.813	2	2	ust	cz042	5341	154.36	010300000001000000BD03000000000040789427C1000000...
2	3316542464	413736.625	3	3	kar	cz041	3317	91.04	010300000001000000B1020000C976BE9FCB2329C1000000...
3	4766501888	553792.438	4	4	krh	cz052	4767	115.97	010300000001000000A0030000C976BE9FEB8424C1000000...
4	496079072	150967.641	5	5	pha	cz011	496	2447.93	01030000000100000021010000C976BE9F7D2A26C1000000...
5	11013148672	1017988.875	6	6	str	cz021	11013	101.05	010300000002000000CC060000C976BE9F0B2028C1000000...
6	7566341120	582346.063	7	7	plz	cz032	7566	73.79	010300000001000000C5030000000000C0A3AE2AC1000000...
7	4523096064	495384.844	8	8	par	cz053	4523	112.47	0103000000010000006A030000378941602A5D23C1000000...
8	5566198272	540849.125	9	9	ost	cz081	5566	230.56	010300000001000000F03000000000000E8331FC1000000...
9	6924478976	577241.25	10	10	jih	cz061	6924	75.26	010300000001000000590400003789416039FB22C1000000...
10	5141526528	642541.25	11	11	olo	cz071	5142	125.01	01030000000100000035040000C976BE9F6BAC21C1000000...
11	3961177856	386157.625	12	12	zln	cz072	3961	150.7	010300000001000000460200000000040B40721C1000000...
12	7065391104	762564.875	13	13	bm	cz062	7065	160.91	0103000000010000005F05000000000040861B21C1000000...
13	10070537216	767645.938	14	14	bud	cz031	10071	61.85	0103000000010000006E0500000000040B2FE24C1000000...



Příklad tabulky

- ◆ CREATE TABLE mereni(id int4, name VARCHAR(128));
- ◆ SELECT AddGeometryColumn('public', '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));
 - ◆ **Otázka: je efektivní takto zapisovat data?**
- ◆ SELECT id, name, the_geom FROM mereni;

Příklad

- ◆ `SELECT id, name, the_geom FROM mereni;`
- ◆ `id | name | the_geom`
- ◆ `--+-----`
- ◆ `1 | Bod c. 1 | 0101000020E610000021B07268910D3240D578E9263`

- ◆ SELECT id, name, AsText(the_geom) from mereni;
- ◆
- ◆ id | name | astext
- ◆ ----+-----+-----
- ◆ 1 | Bod c. 1 | POINT(18.053 49.689)

Vytvoření tabulky

- ◆ Vytvoření tabulky (fakticky atributová část)
- ◆ Zadání nového sloupce s prostorovým rozšířením (pomocí funkce postgisu)
- ◆ Vkládání dat....
- ◆ Dotazy
 - ◆ Klíčový problém
 - ◆ U geodat je typické jednou vytvořit soubor, občas provést malou editace **a často se ptát**

Import dat

- ◆ Data import/export by utility shp2pgsql and pgsql2shp:
- ◆ `shp2pgsql -s 4326 mereni mereni | psql -U username -h localhost -d gdb`
- ◆ `pgsql2shp -f mereni -h localhost -u username gdb mereni`
- ◆ Data import/export by utility ogr2ogr:
- ◆ `ogr2ogr -f PostgreSQL PG:'dbname=gdb user=username' mereni.shp`
- ◆ `ogr2ogr -f "MapInfo File" mereni PG:'dbname=gdb user=username' -sql "SELECT * FROM mereni"`

Prostorové dotazy

- ◆ `SELECT h.name, p.name FROM hospitals h, pubs p WHERE Distance(h.the_geom, p.the_geom) < 250;`
- ◆ `SELECT nazev, y(Transform(the_geom,4326)) AS latitude FROM mesta_cr ORDER BY latitude ASC LIMIT 1;`
- ◆ `SELECT * from mtable where the_geom && 'POLYGON(....)';`

Dotazy

- ◆ suma délky všech silnic
 - ◆ `SELECT sum(length(the_geom))/1000 AS km_roads FROM bc_roads;`
- ◆ plocha vybraného města
 - ◆ `SELECT area(the_geom)/10000 AS hectares FROM bc_municipality WHERE name = 'PRINCE GEORGE';`
- ◆ největší město (plochou)
 - ◆ `SELECT name, area(the_geom)/10000 AS hectares`
 - ◆ `FROM bc_municipality`
 - ◆ `ORDER BY hectares DESC`
 - ◆ `LIMIT 1;`

- ◆ SELECT m.name,
sum(lengthr.the_geom)/1000 as roads_km
- ◆ FROM bc_roads AS r, bc_municipality AS m
- ◆ WHERE r.the_geom && m.the_geom
- ◆ AND contains(m.the_geom, r.the_geom)
- ◆ GROUP BY m.name
- ◆ ORDER BY roads_km;

Výstup do MapServeru

- ◆ LAYER
- ◆ CONNECTIONTYPE postgis
- ◆ NAME "widehighways"
- ◆ CONNECTION "user=dbuser
dbname=gisdatabase host=bigserver"
- ◆ DATA "geom from roads"
- ◆ STATUS ON
- ◆ TYPE LINE
- ◆ FILTER "type = 'highway' and numlanes >=
4"

Další navazující přednášky

- ◆ Vstup údajů
- ◆ Analýza a syntéza údajů
- ◆ Vizualizace údajů