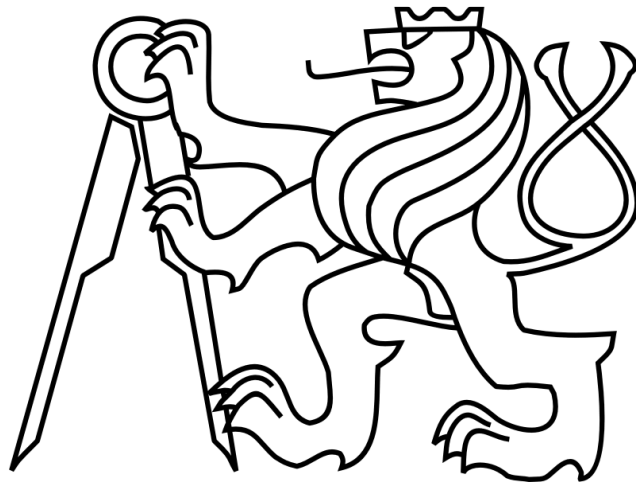


ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

FAKULTA STAVEBNÍ

OBOR GEODÉZIE A KARTOGRAFIE



BAKALÁŘSKÁ PRÁCE

DATABÁZOVÉ ULOŽENÍ PROSTOROVÝCH DAT POMOCÍ POSTGIS

V Praze, 2008

Markéta Havlíčková

## Prohlášení

Prohlašuji, že jsem svou bakalářskou práci vypracovala samostatně a použila jsem pouze podklady uvedené v příloženém seznamu.

V Praze dne:

Podpis:

## Anotace

Práce se zabývá možnostmi ukládání prostorových dat do relačního databázového systému PostgreSQL a jeho rozšíření PostGIS. Dále jsou popsány předdefinované funkce k analýze dat, které jsou v PostGIS obsaženy a jsou uvedeny jejich praktické příklady. Práce se také věnuje problému Křovákova zobrazení.

Klíčová slova: PostGIS, PostgreSQL, prostorová data

## Abstract

My work is focused on possibilities of spatial data storing to relation database system PostgreSQL and its extension PostGIS. There are described data analyst functions, which are contained in PostGIS, with practical examples. The work deals with problem of Krovak's projection.

Key words: PostGIS, PostgreSQL, spatial data

## Obsah

<b>1</b>	<b>Terminologie, zkratky .....</b>	<b>1</b>
<b>2</b>	<b>Úvod .....</b>	<b>2</b>
<b>3</b>	<b>Možnosti ukládání geografických dat .....</b>	<b>3</b>
3.1	Souborové ukládání geografických dat.....	3
3.1.1	Výhody souborového ukládání dat .....	3
3.1.2	Nevýhody souborového ukládání dat.....	4
3.2	Ukládání geografických dat do databáze .....	4
3.2.1	Výhody databázového ukládání dat .....	4
3.2.2	Nevýhody databázového ukládání dat .....	4
<b>4</b>	<b>SŘBD umožňující ukládání prostorových dat .....</b>	<b>5</b>
4.1	Komerční .....	5
4.1.1	Oracle Locator, Oracle Spatial.....	5
4.2	Open source .....	5
4.2.1	PostgreSQL/PostGIS .....	5
<b>5</b>	<b>Instalace PostgreSQL a PostGIS .....</b>	<b>7</b>
<b>6</b>	<b>Objekty databáze PostGIS.....</b>	<b>13</b>
6.1	Základní typy geografických dat .....	13
6.2	Předdefinované tabulky .....	14
6.2.1	Tabulka GEOMETRY_COLUMNS .....	14
6.2.2	Tabulka SPATIAL_REF_SYS .....	15
6.3	Tabulka s prostorovými daty .....	16
<b>7</b>	<b>Import dat do PostGIS .....</b>	<b>17</b>
7.1	Pomocí SQL souboru.....	17
7.2	Pomocí zaváděcího programu shp2pgsql.....	17
7.3	Pomocí prohlížeče geografických dat.....	18
<b>8</b>	<b>Zobrazení geografických dat .....</b>	<b>19</b>

8.1	ArcGIS .....	19
8.2	Quantum GIS .....	20
8.3	GRASS .....	21
<b>9</b>	<b>Referenční systémy – problém Křovákova zobrazení .....</b>	<b>22</b>
9.1	Křovákovo zobrazení.....	22
9.2	Transformace .....	23
<b>10</b>	<b>Prostorové funkce PostGIS .....</b>	<b>25</b>
10.1	Funkce pro analýzu dat.....	25
10.1.1	Výběrové funkce.....	25
10.1.2	Výpočtové funkce .....	25
10.1.3	Analytické funkce .....	26
<b>11</b>	<b>Praktické příklady prostorových funkcí.....</b>	<b>28</b>
11.1	Výběrové funkce.....	29
11.2	Výpočtové funkce .....	32
11.2.1	Funkce SUM .....	32
11.2.2	Funkce MIN .....	32
11.3	Měřicí funkce.....	33
11.3.1	Funkce výměra.....	33
11.3.2	Funkce obvod.....	33
11.3.3	Funkce vzdálenost.....	34
11.3.4	Funkce generalizace.....	34
11.4	Funkce překrytí.....	36
11.4.1	Funkce průnik .....	36
11.5	Funkce v okolí .....	37
11.5.1	Vyhledávací funkce .....	37
11.5.2	Funkce blízkosti.....	38
<b>12</b>	<b>Závěr .....</b>	<b>40</b>
<b>13</b>	<b>Seznam použité literatury .....</b>	<b>41</b>

## 1 Terminologie, zkratky

Open source GIS – volně šiřitelný software s přístupným zdrojovým kódem

PostgreSQL – relační databázový systém s volně přístupným zdrojovým kódem

PostGIS – rozšíření PostgreSQL, podporující ukládání, správu prostorových geografických dat a využívání prostorových funkcí přímo nad databázemi.

WKT- Well-known text – formát, kterým je reprezentována vektorová geometrie objektů na mapách, prostorové referenční systémy a transformace mezi nimi.

WKB – Well-known binary - binární formát ukládání geografických dat v PostGIS

SŘBD – Systém řízení báze dat

GNU GPL – General Public License – licence pro svobodný software, která spolu s licencí GFDL tvoří základ projektu GNU. Zdrojové kódy tohoto softwaru pod GPL mohou být svobodně užívány, upravovány, dále však musí být šířeny znovu pod GPL.

EPSG – European Petroleum Survey Group (1986 - 2005) vědecká organizace, která sestavila EPSG Geodetic Parameter Set (sadu geodetických parametrů), definovala číselné identifikátory pro mnoho zobrazení a každému z nich přiřadila definici zobrazení a metadata o souřadnicích (nultý poledník, jednotky apod.)

## 2 Úvod

Ve výuce oboru geodézie a kartografie se setkávám s geografickými informačními systémy, které uživatelům umožňují práci s mapovými podklady a geografickými daty, především tedy s programem ArcGIS. Jde o uživatelsky příjemné prostředí s velkým množstvím funkcí. Data, se kterými tento GIS pracuje, jsou většinou souborově uložena. Proto, když jsem si vybírala téma bakalářské práce, mě zaujala myšlenka, že je možné data ukládat i do relačních databází s rozšířením pro prostorová data.

Před dvěma lety vznikl na ČVUT obor geoinformatika, který kombinuje studium geodézie a informatiky, a jehož principy v podstatě splňuje téma prostorových databází. Kombinuje se zde práce s geografickými daty s nutností znát fungování databází a programovací jazyky (např. SQL). V současné době na naší fakultě zatím neexistuje předmět, který by se touto problematikou zabýval, proto bych ráda vytvořila práci, která by v budoucnu mohla být úvodem do předmětu s takovouto tematikou.

Cílem této bakalářské práce je především seznámení s prací s relačním databázovým systémem s otevřeným kódem PostgreSQL a jeho rozšíření PostGIS umožňující ukládání a práci s prostorovými daty.

První část této práce bude obsahovat především teoretický úvod, tedy jaké jsou vůbec možnosti ukládání prostorových dat, které databázové systémy umožňují práci s těmito daty, možnosti prohlížení a zobrazování geografických dat a také bližší popis PostgreSQL/PostGIS a jeho objektů.

Druhá část se bude zabývat především praktickým užitím analytických funkcí, pomocí nichž je možné s daty pracovat a které PostGIS primárně obsahuje, a také referenčními systémy, tedy především Křovákovým zobrazením, užívající se na území České republiky.

## 3 Možnosti ukládání geografických dat

V zásadě existují dvě možnosti ukládání geografických dat a to souborové ukládání a ukládání dat do databáze. V této kapitole budou tyto možnosti zhodnoceny.

### 3.1 Souborové ukládání geografických dat

Příkladem souborového ukládání dat může být například nejrozšířenější vektorový formát shapefile. Shapefile byl vyvinut firmou ESRI jako možnost přenositelnosti dat mezi softwarem ESRI a dalšími softwarovými produkty. Shapefile se skládá z několika souborů, povinných a nepovinných, se stejným názvem ale odlišnou příponou např. jezera.shp, jezera.shx a jezera.dbf.

*Povinnými soubory jsou:*

SHP – obsahuje vlastní geometrii

SHX – shape index format - obsahuje polohové indexy, které umožňují rychlejší vyhledávání prvků

DBF – obsahuje tabulky atributů ve formátu dBase III

*Nepovinnými soubory jsou:*

PRJ – projection format – obsahuje definici kartografického zobrazení a souřadnicového systému v jazyku WKT.

Dalšími příkladem souborového ukládání dat je formát DGN, který je podporován v produktech firmy Bentley Systems – tvůrce programu Microstation. Tento formát existuje ve dvou verzích: DGN V7 a DGN V8, který je novější a byla u něj odstraněna omezení z formátu DGN V7. Příkladem změny je např. maximální velikost výkresového souboru ze 32MB ve verzi DGN V7 na 4GB ve verzi DGN V8.

DWG je primárním formátem programu AutoCAD, firmy Autodesk. Výměnnou variantou je textový formát DXF.

#### 3.1.1 Výhody souborového ukládání dat

Soubory jsou jednoduše přenositelné, jsou podporovány většinou software.

Pro většinu běžných uživatelů je souborové ukládání jednodušší, než práce na úrovni databáze, k níž je potřeba znát její fungování a správu.



Většina prostorových dat je na úřadech k dostání právě v souborovém formátu, čili jejich výhodou je dostupnost.

### 3.1.2 Nevýhody souborového ukládání dat

Každý shapefile může teoreticky obsahovat několik typů objektů (polygon, point). Takové ukládání ale GIS nepodporují, a proto se tyto geometrické typy používají odděleně. Tzn. jezera(polygon) a řeky(linestring) budou ve dvou shapefiles.

Propojování mezi vrstvami je složitější než v databázi, záleží na softwaru.

Další nevýhodou souborového ukládání je relativně pomalé vyhledávání, které je sice urychlováno prostorovými indexy, ale nikdy nebude tak rychlé jako v databázi.

## 3.2 Ukládání geografických dat do databáze

Prostorová data je možné ukládat do tzv. prostorových databází, které jsou založené na principu relační databáze, do které jsou přidány prostorové atributy. Do tabulky s daty o objektu je vložen zvláštní sloupec `the_geom`, který obsahuje geometrii objektu.

### 3.2.1 Výhody databázového ukládání dat

Prostorové databáze přejímají výhody běžných relačních databází. To například znamená, že data mohou být uchovávána na serveru a je k nim umožněn poměrně jednoduchý vzdálený přístup, s daty může pracovat více uživatelů najednou a nemusí být kopírována na lokální počítač.

Velkou výhodou je uložení veškerých dat přehledně na jednom místě (souboru, adresáři).

Informace o souřadnicovém systému jsou uloženy přímo v databázi.

Díky jednoznačným identifikátorům lze poměrně jednoduše dohledat konkrétní data.

### 3.2.2 Nevýhody databázového ukládání dat

Pro běžné uživatele je databázové uložení dat a práce s nimi složitější. Při práci s databází je nutná znalost např. jazyka SQL a principů databáze. Ve většině případů by na údržbu databáze byl potřeba další zaměstnanec – správce databáze. Proto se zavedení prostorové databáze vyplatí hlavně u velkého množství geografických dat.

## 4 SŘBD umožňující ukládání prostorových dat

### 4.1 Komerční

#### 4.1.1 Oracle Locator, Oracle Spatial

Součástí každé databáze Oracle Database je Oracle Locator umožňující ukládání dat a základní dotazy, jde o základní verzi. Oracle Spatial, obsažený v balíčku Oracle Database 11g Enterprise Edition, podporuje správu všech geografických dat včetně vektorových a rastrových formátů, topologii a síťové modely.

Ceny licencí těchto databází se pohybují v řádu stovek tisíc korun za procesor. Výjimkou je Oracle Database 10g Express Edition, která je poskytována zdarma včetně komponenty Oracle Locator, ale s jistými omezeními – lze ji na příklad využít pro správu maximálně 4GB dat.

### 4.2 Open source

Hlavní výhodou open source produktů je nepochybně nulová pořizovací cena. I přesto se ale jedná o kvalitní software. Rozdíl mezi komerčním a open source produktem je v tom, že open source, jak už název napovídá, má otevřený zdrojový kód. To znamená, že k němu má přístup rozsáhlý tým programátorů, kteří jsou roztroušeni po celém světě a vzhledem k nulové pořizovací ceně také velké množství uživatelů, kteří jsou schopni odhalit nedokonalosti a chyby softwaru při jeho každodenním používání. Vzniká tak jakási vazba mezi programátory a uživateli, díky které mohou programátoři relativně rychle reagovat a daný problém odstranit.

Mezi nejvýznamnější open source databázové produkty patří PostgreSQL a MySQL, oba tyto produkty umožňují ukládání prostorových dat.

#### 4.2.1 PostgreSQL/PostGIS

Databázový systém PostgreSQL je založen na principu relační databáze. Je podporován všemi hlavními operačními systémy jako je Linux, UNIX a Windows. Tento open source produkt je vyvíjen a vylepšován více než 15 let. Kromě běžných datových typů jako je integer, boolean, char, varchar apod., podporuje také ukládání binárních objektů včetně obrázků, audia a videa.

---

PostgreSQL je rozšířen o komponentu PostGIS umožňující ukládání a práci s prostorovými daty. Díky tomu může být PostgreSQL server použit jako databáze pro geografické informační systémy.

PostGIS se řídí normou „Simple Features Specification for SQL“.

## 5 Instalace PostgreSQL a PostGIS

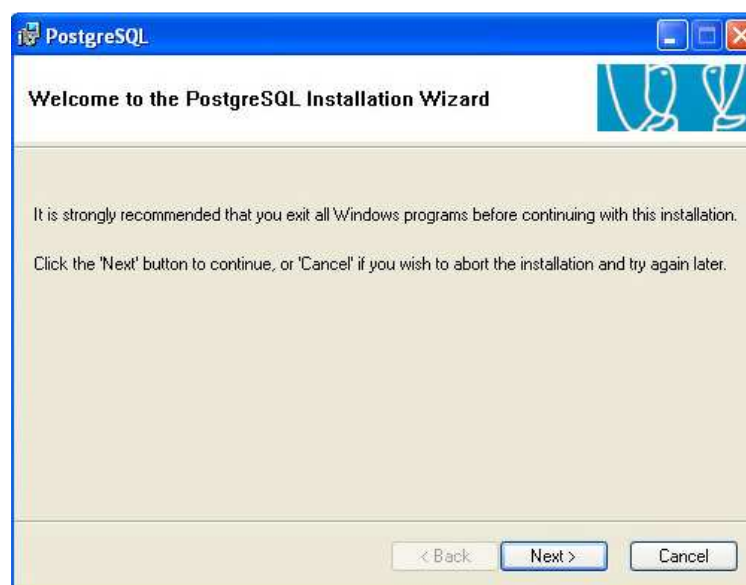
Tato kapitola obsahuje návod k instalaci PostgreSQL a jeho rozšíření PostGIS pro OS Windows.

- a) Nejprve byla stažena nejnovější volně dostupná verze postgresql-8.2.5-1.zip pro Windows. Poté byl archiv rozbalen a byl spuštěn soubor postgresql-8.2.msi.
- b) Na první obrazovce je možné vybrat uživatelský jazyk instalace.



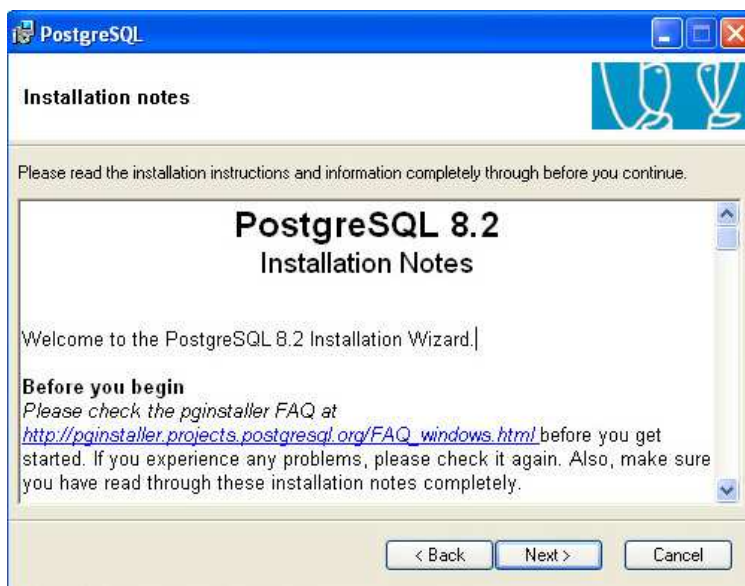
Obrázek 5-1 Instalace PostgreSQL

- c) Druhá obrazovka nás pouze uvítá v průvodci instalací.



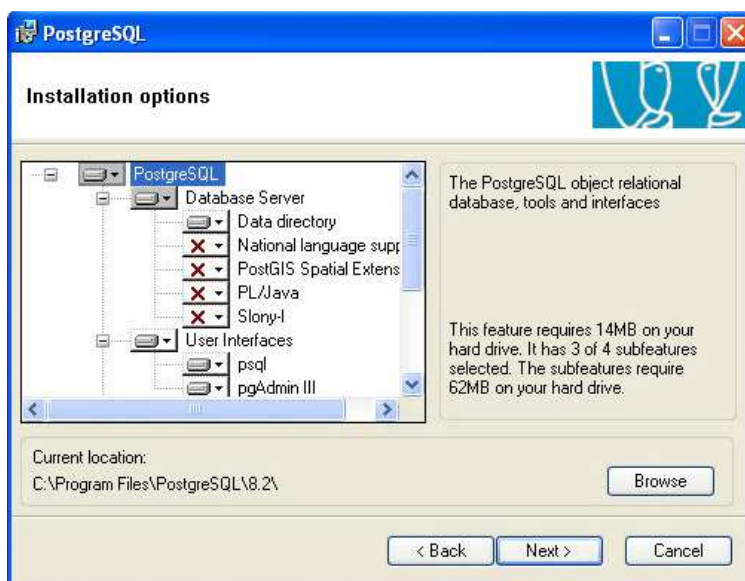
Obrázek 5-2 Instalace PostgreSQL

- d) Třetí obrazovka upozorňuje na přečtení poznámek k instalaci.



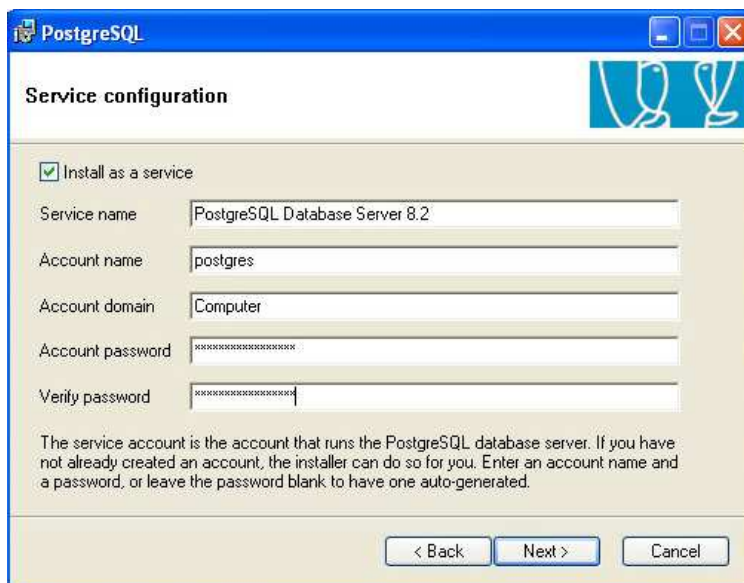
Obrázek 5-3 Instalace PostgreSQL

- e) Na čtvrté obrazovce je nutné vybrat National language a zvolit první možnost, dále kliknout na PostgreSQL a vybrat umístění instalace na HDD.



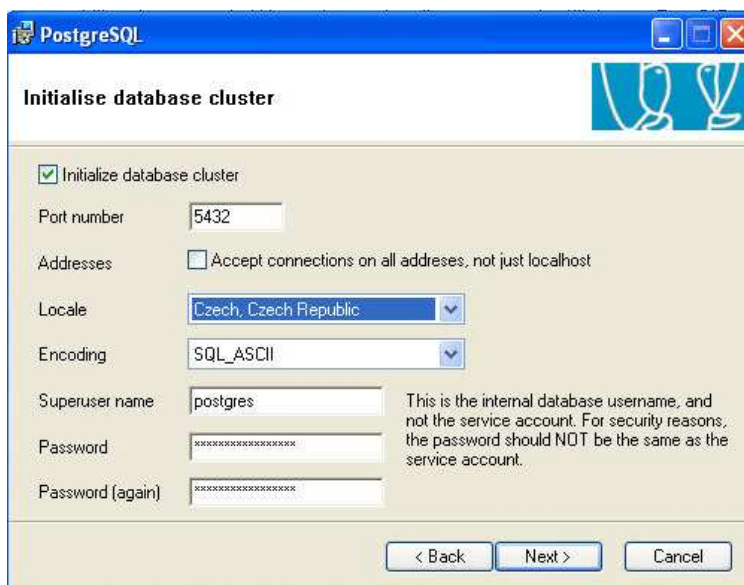
Obrázek 5-4 Instalace PostgreSQL

- f) Na páté obrazovce je vytvářen uživatelský účet se jménem postgres, je zadáno heslo a doména



Obrázek 5-5 Instalace PostgreSQL

- g) Na šesté obrazovce je nastavován databázový systém, číslo portu je standardně 5432. Dále je možné určit dostupnost databáze z jiného počítače zaškrtnutím kolonky Addresses. Na výběr je dále kódování – doporučuji UTF8, dále se vyplňuje heslo, pro přihlášení k databázi.



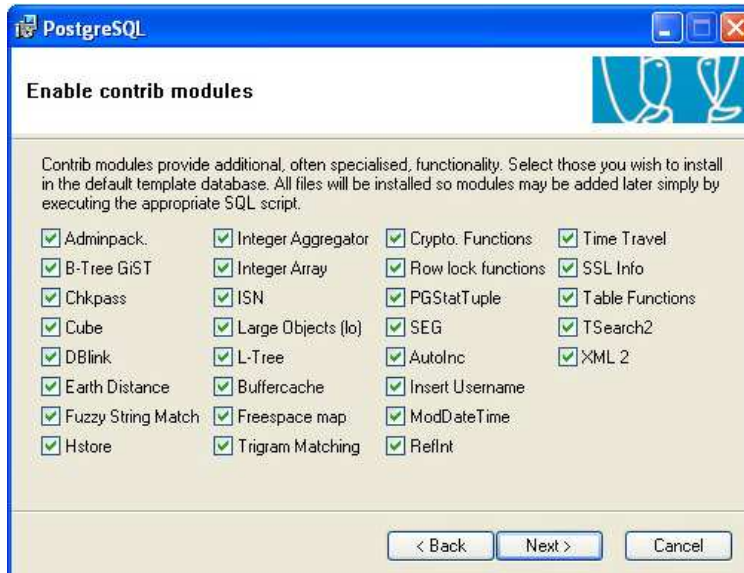
Obrázek 5-6 Instalace PostgreSQL

- h) Na sedmé obrazovce je možné vybrat procedurální jazyky, včetně Perlu nebo Pythonu, mně stačilo PL/pgsql.



Obrázek 5-7 Instalace PostgreSQL

- i) Osmá obrazovka nabízí výběr libovolných modulů.



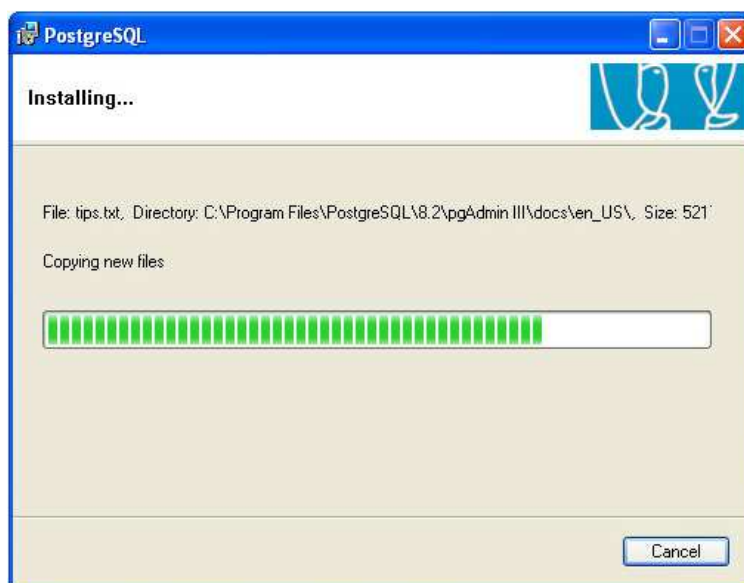
Obrázek 5-8 Instalace PostgreSQL

- j) Devátá obrazovka upozorňuje, že instalace je připravena.



Obrázek 5-9 Instalace PostgreSQL

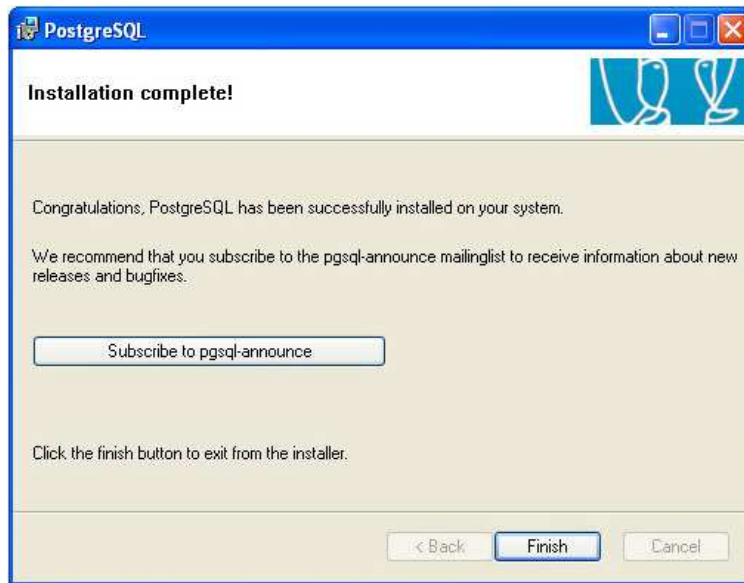
- k) Na desáté obrazovce probíhá samotná instalace.



Obrázek 5-10 Instalace PostgreSQL



- 1) Instalace je kompletní, doporučuje se restartovat počítač.



*Obrázek 5-11 Instalace PostgreSQL*

Samotná databáze PostgreSQL se ovládá pomocí příkazové řádky. Pro příjemnější práci s daty proto doporučuji nainstalovat také program PgAdmin III. Součástí instalace PostgreSQL je sice verze PgAdmin III (viz obrázek 5-4), ten však nespolupracuje dobře s Windows, a proto jsem si PgAdmin III stáhla a nainstalovala zvlášť. Instalace PgAdmin je poměrně jednoduchá, proto není nutné ji zde zmiňovat.

PostGIS je možné užívat také na platformě Linux, pro úplnost uvádím také odkaz na instalaci pro tento operační systém [4].

## 6 Objekty databáze PostGIS

V této kapitole budou definovány a objasněny základní objekty v databázi PostGIS. Objekty a funkce PostGIS se řídí standardem OpenGIS Simple Features Specification for SQL, který specifikuje typy geometrie (`point`, `multipoint`, `linestring` atd.) a jejich vztahy. Definuje a objasňuje funkce pro analýzu geografických dat. Jeho součástí je také podrobný popis architektury databáze - předdefinovaná tabulka obsahující definice referenčních systémů `SPATIAL_REF_SYS` a tabulka `GEOMETRY_COLUMNS`, která obsahuje souhrnné informace o datových tabulkách v databázi.

Cílem této práce je popsat nejběžněji používané funkce (kap. 10), základní typy geografických dat a popsat výše uvedené předdefinované tabulky.

Samotný standard je velice obsáhlý a je možné si ho prohlédnout v originále na webových stránkách Open Geospatial Consortium, Inc. [6].

Na toto téma byla také v češtině zpracována diplomová práce Topologické operace v geografických informačních systémech [7], jejíž první část je věnována právě tomuto standardu.

### 6.1 Základní typy geografických dat

Prvek typu Bod vyjadřuje objekt, který je tak malý, že ho nelze vyjádřit linií nebo plochou. Příkladem může být kóta. Jako `MULTIPOINT` je uvažován shluk bodů, který je do databáze uložen jako jeden objekt.

```
POINT (0 0),  
MULTIPOINT (0 0,1 2)
```

Prvek typu Linie vyjadřuje objekt, který je protáhlého tvaru, ale je tak úzký, že ho nelze vyjádřit polygonem. Příkladem je řeka, komunikace. `MULTILINESTRING` je objekt skládající se z několika linií, opět je v databázi uložen jako jeden objekt.

```
LINestring (0 0,1 1,1 2)  
MULTILINESTRING ((0 0,1 1,1 2),(2 3,3 2,5 4))
```

Prvek typu Polygon vyjadřuje objekt, který lze označit jako souvislou plochu. Např. porosty, průmyslové areály. `MULTIPOLYGON` je soubor několika polygonů.

```
POLYGON ((0 0,4 0,4 4,0 4,0 0),(1 1, 2 1, 2 2, 1 2,1 1))
MULTIPOLYGON (((0 0,4 0,4 4,0 4,0 0),(1 1,2 1,2 2,1 2,1
1)),((-1 -1,-1 -2,-2 -2,-2 1,-1 -1)))
```

Geometrycollection se skládá z různých typů geometrie, je uložena jako jeden objekt.

```
GEOMETRYCOLLECTION(POINT(2 3),LINESTRING((2 3,3 4)))
```

## 6.2 Předdefinované tabulky

Prostorová databáze obsahuje v zásadě dva druhy tabulek. Dvě souhrnné tabulky GEOMETRY\_COLUMNS a SPATIAL\_REF\_SYS a tabulky prostorových objektů, které se od běžných tabulek v relačních databázích liší tím, že obsahují sloupec, ve kterém je charakterizována geometrie.

### 6.2.1 Tabulka GEOMETRY\_COLUMNS

Tato souhrnná tabulka obsahuje metadata o tabulkách s geografickými daty a obsahuje tyto sloupce:

F\_TABLE\_CATALOG – názvy katalogů. Termíny „catalog“ a „schema“, pocházejí z databáze Oracle, PostGIS nemá pro „catalog“ analogii a proto zůstává tento sloupec prázdný.

F\_TABLE\_SCHEMA – schéma je kolekce databázových objektů – triggerů, uživatelů, tabulek apod. Databázová schémata vytváří správce serveru. Každý uživatel má k dispozici svoje schéma s vlastním přihlašovacím jménem a heslem.

F\_TABLE\_NAME – názvy tabulek

F\_GEOMETRY\_COLUMN – názvy geometrických sloupců všech tabulek v databázi

COORD\_DIMENSION – určuje dimenzi souřadnic

SRID – identifikátor – cizí klíč odkazující se na tabulku SPATIAL\_REF\_SYS na konkrétní referenční systém

TYPE – obsahuje typ geometrie v jednotlivých tabulkách (např. LINESTRING)

#### Příklad 6-1 SQL příkaz k vytvoření tabulky GEOMETRY\_COLUMNS

```
CREATE TABLE geometry_columns (
  f_table_catalog  VARCHAR(256) NOT NULL,
  f_table_schema   VARCHAR(256) NOT NULL,
  f_table_name     VARCHAR(256) NOT NULL,
  f_geometry_column VARCHAR(256) NOT NULL,
  coord_dimension  INTEGER NOT NULL,
```

```
srid          INTEGER NOT NULL,
type         VARCHAR(30) NOT NULL )
```

### 6.2.2 Tabulka SPATIAL\_REF\_SYS

Tato tabulka obsahuje informace o referenčních systémech a jejich definice a skládá se z těchto sloupců:

SRID – identifikátor

AUTH\_SRID – identifikátor autority

AUTH\_NAME – jméno autority, která definovala referenční systém např. EPSG

SRTEXT – WKT popis referenčního systému.

#### Příklad 6-2 Zápis SRTEXT

```
PROJCS["NAD83 / UTM Zone 10N",
  GEOGCS["NAD83",
    DATUM["North_American_Datum_1983",
      SPHEROID["GRS 1980",6378137,298.257222101]
    ],
    PRIMEM["Greenwich",0],
    UNIT["degree",0.0174532925199433] ],
  PROJECTION["Transverse_Mercator"],
  PARAMETER["latitude_of_origin",0],
  PARAMETER["central_meridian",-123],
  PARAMETER["scale_factor",0.9996],
  PARAMETER["false_easting",500000],
  PARAMETER["false_northing",0],
  UNIT["metre",1] ]
```

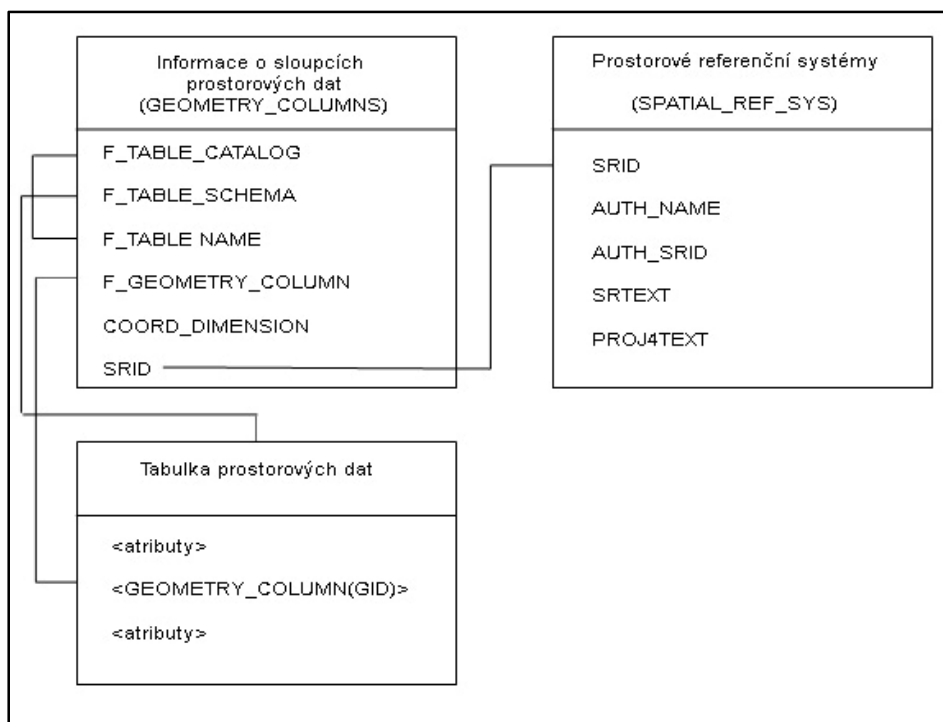
PROJ4TEXT – PostGIS využívá knihovnu Proj k transformaci souřadnic mezi různými systémy. PROJ4TEXT je textová definice souřadnicového systému v Proj.

#### Příklad 6-3 Zápis PROJ4TEXT

```
+proj=utm +zone=10 +ellps=clrk66 +datum=NAD27 +units=m
```

#### Příklad 6-4 SQL příkaz k vytvoření tabulky SPATIAL\_REF\_SYS

```
CREATE TABLE spatial_ref_sys (
  srid          INTEGER NOT NULL PRIMARY KEY,
  auth_name     VARCHAR(256),
  auth_srid     INTEGER,
  srtext        VARCHAR(2048),
  proj4text     VARCHAR(2048))
```



Obrázek 6-1 Relace mezi tabulkami PostGIS a tabulkami s daty

### 6.3 Tabulka s prostorovými daty

Tabulka s prostorovými daty je rovněž vytvořena SQL příkazem. Vytváří se obdobně jako tabulka v „klasické“ relační databázi s tím rozdílem, že v této tabulce je vytvořen pomocí funkce `AddGeometryColumn(character varying, character varying, character varying, integer, character varying, integer)` sloupec s geometrií objektu.

#### Příklad 6-5 Vytvoření tabulky s daty

```
CREATE TABLE "public"."lesycr"
  (gid serial PRIMARY KEY,
  "area" float8,
  "perimeter" float8,
  "lesy_" int8,
  "lesy_id" int8);
SELECT AddGeometryColumn('public','lesycr','the_geom','-1',
'MULTIPOLYGON',2);
INSERT INTO "public"."lesycr"
("area","perimeter","lesy_","lesy_id",the_geom)
VALUES
('6344148.500','16521.615','2','1','010600000001000000010);
INSERT INTO "public"."lesycr"
("area","perimeter","lesy_","lesy_id",the_geom) VALUES
('15583692.000','36834.559','3','1','01060000000100000001030
0000001000004);
```

## 7 Import dat do PostGIS

Prostorových dat je většinou velké množství a není možné je do databáze zadávat pouze „ručně“ pomocí SQL. Proto se většinou využívá jedna z následujících možností:

### 7.1 Pomocí SQL souboru

SQL soubor pro import dat do tabulky může vypadat například takto

#### Příklad 7-1 SQL soubor

```
BEGIN
INSERT INTO
"public"."silnicedalnice"("id","nazev","stavobjekt","stavobjekt1",the_geom)
VALUES ('12749','446','026','Objekt v provozu','19837287429');
INSERT INTO
"public"."silnicedalnice"("id","nazev","stavobjekt","stavobjekt1",
the_geom)
VALUES ('11033','446','026','Objekt v
provozu','2849758392');
END;
```

### 7.2 Pomocí zaváděcího programu shp2pgsql

Pokud není k dispozici SQL soubor, což bude ve většině případů, je možné importovat data do PostGIS za použití zaváděcího programu shp2pgsql.exe, který je součástí instalace PostgreSQL. Tato aplikace vygeneruje z SHP souboru SQL soubor, kterým se posléze vytvoří a naplní tabulka v databázi.

Tato aplikace se spouští z příkazové řádky např. pomocí následujícího příkazu:

#### Příklad 7-2 Příkaz shp2pgsql

```
shp2pgsql roads.shp schema.table > roads.sql
psql -d roadsdb -f roads.sql
```

### **7.3 Pomocí prohlížeče geografických dat**

Další možností je importovat nebo exportovat dat pomocí programů pro prohlížení geografických dat uložených v databázi. Postup při tomto importu záleží na typu prohlížeče. Více informací o prohlížečích a jejich možnostech je uvedeno v kapitole 8.

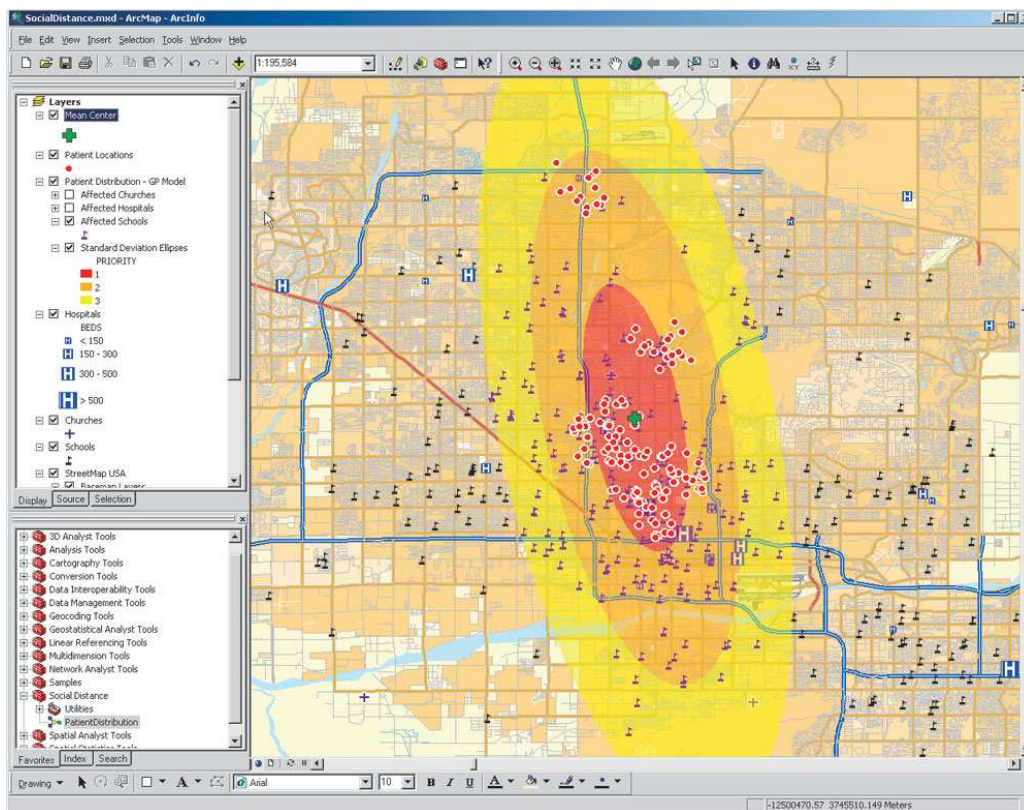
## 8 Zobrazení geografických dat

Vzhledem k tomu, že data se v databázi zobrazují jen v číselných hodnotách je pro lepší představu o datech potřeba zobrazit je i graficky. K tomu slouží nejrůznější ať už komerční nebo open source prohlížeče. Takových programů, které mohou data nejen zobrazovat, ale i pracovat s vrstvami vytvořenými tabulkami v PostGIS, je celá řada, nicméně některé jsou stále ve fázi vývoje. V této kapitole se budu zabývat těmi programy, které jsou již poměrně dobře vybaveny.

### 8.1 ArcGIS

Rozšířeným produktem pro platformu Windows je ArcGIS od firmy ESRI. Program ArcGIS je podle druhu licence rozlišen na několik různých stupňů, které jsou od sebe odlišeny různou šíří funkcí. V následujícím výčtu jsou programy ArcGIS podle těchto stupňů seřazeny:

- ArcReader – freeware umožňující pouze základní prohlížení map
- ArcView – software již krytý licenci. Jeho pomocí je možné data nejen prohlížet,



Obrázek 8-1 Ukázka obrazovky ArcGIS (ArcInfo), zdroj [10]



ale už je i editovat a pracovat s daty relační databáze.

- ArcEditor - umožňuje pokročilou editaci dat
- ArcInfo – v něm je možné data i modelovat a různě analyzovat

Tento software obsahuje i další komponenty jako je ArcMap, ArcCatalog, ArcToolbox a ArcGlobe.

ArcGIS podporuje celou řadu formátů (vektorové, rastrové) včetně importu a exportu dat PostGIS. Vrstvu PostGIS je možné přidat pomocí komponenty ArcCatalog.

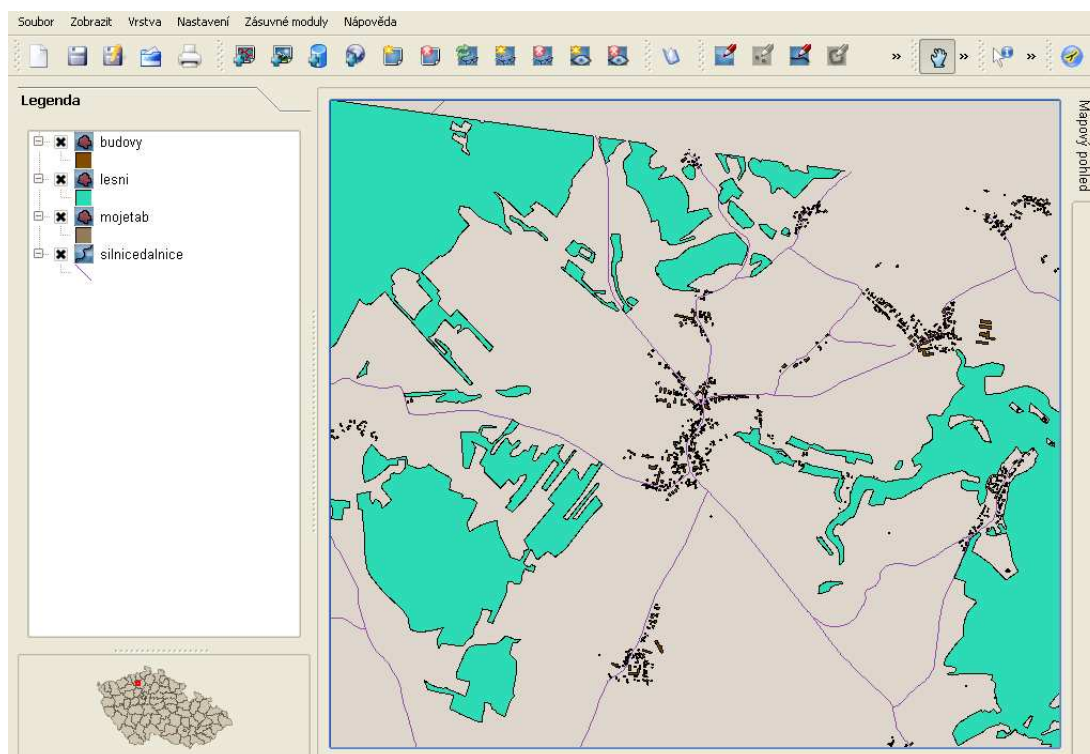
## 8.2 Quantum GIS

Quantum GIS (nebo také QGIS) je open source GIS, který podporuje rastrové, vektorové formáty, včetně prostorových jako je shapefile nebo GeoTIFF, i databázové formáty. Umí využívat webové mapové služby. Pomocí pluginů je možné zobrazovat i data z GPS, pracovat s vrstvami GRASS, importovat data z formátu shapefile do PostgreSQL/PostGIS. Další pluginy je možné naprogramovat pomocí jazyku Python nebo C++.

QGIS je možné používat na různých platformách např. Unix, Windows, Mac OS X.

Výhodou tohoto programu je, že je možné sehnat jeho českou verzi.

Je šířen pod licencí GNU GPL.



Obrázek 8-2 Ukázka obrazovky Quantum GIS

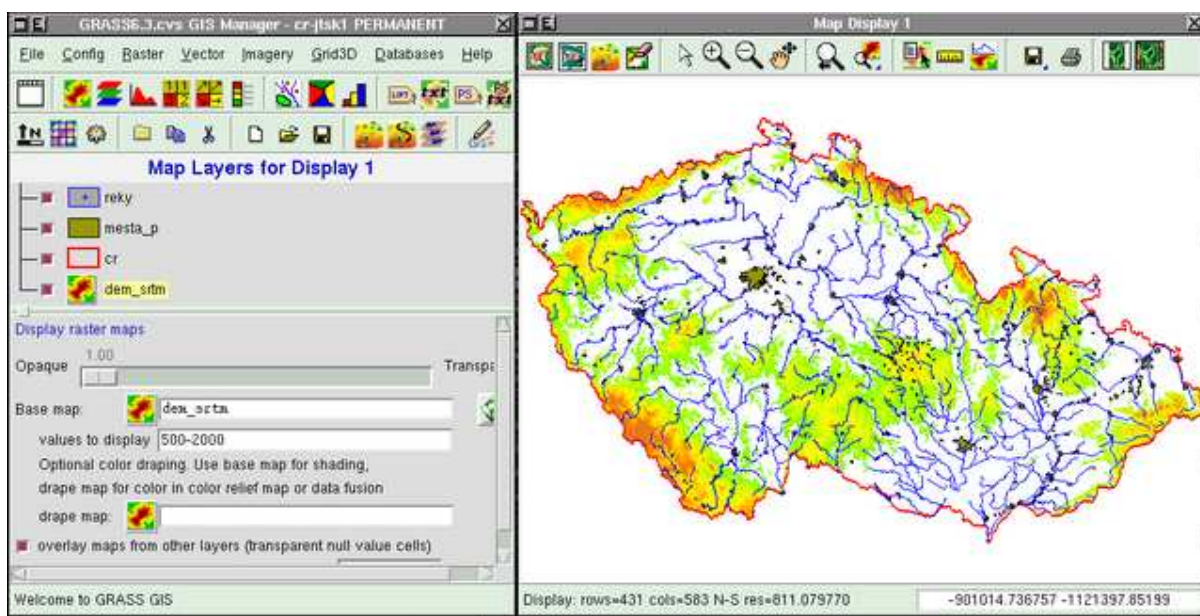
### 8.3 GRASS

Software GRASS je využíván k analýzám prostorových dat, k produkci map, prostorového modelování a vizualizaci, umožňuje práci s vektorovými a rastrovými daty. Od verze 6.0 je v GRASS implementována podpora pro import a export dat PostGIS.

Zkratka GRASS znamená Geographic Resources Analysis Support System – systém podporující analýzu geografických zdrojů. Jde o software šířený pod licencí GNU GPL.

GRASS je určen pro platformy GNU Linux, Mac OS X. Je možné ho nainstalovat i na Windows pomocí Linux-like prostředí cygwin.

GRASS je oficiální projekt Open Source Geospatial Foundation.



Obrázek 8-3 Ukázka obrazovky GRASS, zdroj [5]

## 9 Referenční systémy – problém Křovákova zobrazení

### 9.1 Křovákovo zobrazení

Je žádoucí, aby byl při importu dat přiřazen datům určitý referenční systém, který se vyskytuje v tabulce `SPATIAL_REF_SYS`. Datové tabulce je přiřazen identifikátor `SRID` a tato tabulka je posléze zapsána do tabulky `GEOMETRY_COLUMNS`. Poté je možné provádět transformace mezi různými souřadnicovými systémy.

Křovákovo zobrazení je dvojité konformní kuželové zobrazení v obecné poloze, pojmenováno je po svém autorovi Ing. Josefu Křovákovi. Výchozí referenční plochou je Besselův elipsoid.

Problémem při práci s daty v Křovákově zobrazení je skutečnost, že pracuje s geodetickým systémem os `X`, `Y`, kdy osa `X` je kladná v jižním směru, zatímco osa `Y` je kladná ve směru západním. Běžné geografické informační systémy pracují s matematickým systémem os, tedy osa `X` je kladná ve východním směru a osa `Y` je kladná v severním směru.

Definice Křovákova zobrazení proto existuje hned několik. Primárně je v PostGIS vložená definice, jejíž `SRID` je 2065. Autorem této definice je EPSG a počátek je v tomto případě uveden od Ferra s kladnými geodetickými osami. Podle kartografie jde o správnou definici Křovákova zobrazení.

#### Příklad 9-1 Definice Křovákova zobrazení 2065 v proj

```
+proj=krovak +lat_0=49.5 +lon_0=42.5  
+alpha=30.28813972222222 +k=0.9999 +x_0=0 +y_0=0  
+ellps=bessel +pm=ferro +units=m +no_defs
```

#### Příklad 9-2 Definice Křovákova zobrazení 2065 v srtxt

```
PROJCS["S-JTSK (Ferro) / Krovak",GEOGCS["S-JTSK (Ferro)",  
DATUM["S_JTSK_Ferro",SPHEROID["Bessel  
1841",6377397.155,299.1528128,  
AUTHORITY["EPSG","7004"]],AUTHORITY["EPSG","6818"]],  
PRIMEM["Ferro",-17.66666666666667,AUTHORITY["EPSG","8909"]],  
UNIT["degree",0.01745329251994328,AUTHORITY["EPSG","9122"]],  
AUTHORITY["EPSG","4818"]],  
PROJECTION["Krovak"],  
PARAMETER["latitude_of_center",49.5],  
PARAMETER["longitude_of_center",42.5],  
PARAMETER["azimuth",30.28813972222222],  
PARAMETER["pseudo_standard_parallel_1",78.5],  
PARAMETER["scale_factor",0.9999],PARAMETER["false_easting",0  
],
```

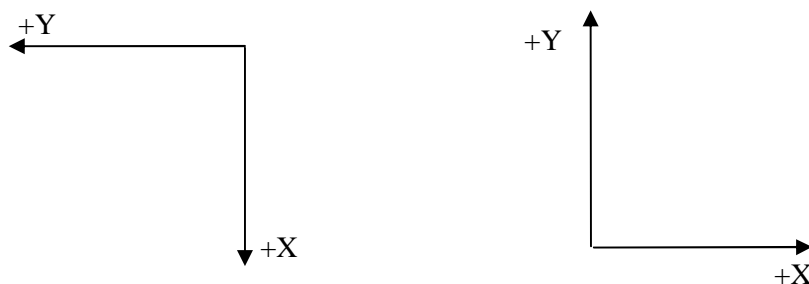
```
PARAMETER["false_northing",0],UNIT["metre",1,AUTHORITY["EPSG",
"9001"]],AUTHORITY["EPSG","2065"]]
```

Avšak vzhledem k tomu, že většina GIS aplikací pracuje s matematickým systémem os, byly nakonec vytvořeny celkem 4 definice tohoto zobrazení v různých kombinacích:

SRID	Počátek	Osy	Autor
2065	Ferro	geodetické osy	EPSG
102065	Greenwich	geodetické osy	ESRI
102066	Ferro	matematické osy	ESRI
102067	Greenwich	matematické osy	ESRI

Geodetické postavení os

Matematické postavení os

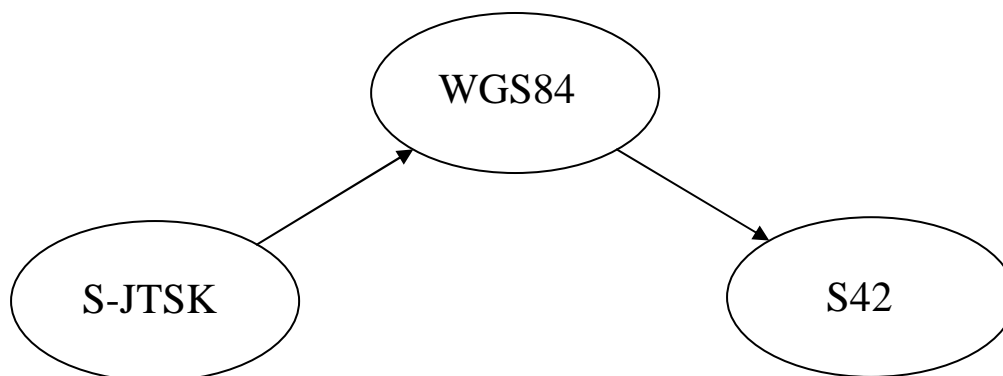


Obrázek 9-1 Rozdíl mezi geodetickým a matematickým systémem os

Dnes je však běžnou praxí používat pro geografická data poslední zmíněnou definici s počátkem v Greenwichi a matematicky položenými osami. Převod mezi definicí 2065 a 102067 je tedy zřejmý  $X = -Y$  a  $Y = -X$ . Proto je žádoucí, aby toto zobrazení bylo do PostGIS doplněno.

## 9.2 Transformace

Transformace mezi jednotlivými souřadnicovými systémy neprobíhá v PostGIS vždy přímo. V PostGIS jsou definovány transformační koeficienty pro transformaci mezi libovolným souřadnicovým systémem a systémem WGS84, proto může transformace vypadat třeba takto:



Obrázek 9-2 Ukázka transformace mezi souřadnicovými systémy S-JTSK a S42

Transformační koeficienty z S-JTSK do WGS84 nejsou konkretizované pro Českou republiku a proto je vhodné tyto transformační koeficienty zpřesnit. Tyto klíče lze zpřesnit pomocí sedmi prvkové Helmertovy transformace, jako v případě [5], kdy bylo použito 174 bodů kampaně DOPNUL, které jsou zaměřeny jak v S-JTSK, tak ve WGS84.

Nové transformační klíče se definují pomocí parametru `+towgs84=570.83789, 85.682641, 462.84673, 4.9984501, 1.5867074, 5.2611106, 3.5610256`

Aby tedy bylo možné korektně transformovat mezi S-JTSK a dalšími systémy, je nutné do PostGIS definovat Křovákovo zobrazení 102067 takto:

### Příklad 9-3 Vložení Křovákova zobrazení 102067 do tabulky `SPATIAL_REF_SYS`

```

INSERT INTO spatial_ref_sys (srid, auth_name, auth_srid,
srtext,
proj4text) VALUES (102067, 'ESRI', 102067,
'PROJCS["SJTSK_Krovak_East_North",GEOGCS["GCS_S_JTSK",DATUM
["D_S_JTSK",SPHEROID["Bessel_1841",6377397.155,299.1528128]]
,
PRIMEM["Greenwich",0],
UNIT["Degree",0.017453292519943295]],PROJECTION["Krovak"],
PARAMETER["False_Easting",0], PARAMETER["False_Northing",0],
PARAMETER["Pseudo_Standard_Parallel_1",78.5],
PARAMETER["Scale_Factor",0.9999],
PARAMETER["Azimuth",30.2881397222222],
PARAMETER["Longitude_Of_Center",24.83333333333333],
PARAMETER["Latitude_Of_Center",49.5], PARAMETER["X_Scale",-
1], PARAMETER["Y_Scale",1],
PARAMETER["XY_Plane_Rotation",90],UNIT["Meter",1]]',
'+proj=krovak +lat_0=49.5 +lon_0=24.83333333333333
+alpha=30.2881397222222 +k=0.9999 +x_0=0 +y_0=0
+ellps=bessel +units=m
+towgs84=570.83789,85.682641,462.84673,4.9984501,1.5867074,5
.2611106,3.5610256');
  
```

## 10 Prostorové funkce PostGIS

Prostorové funkce umožňují práci s prostorovými daty přímo na úrovni databáze. PostGIS nabízí několik set funkcí, nicméně je možné si další funkce naprogramovat např. v Perlu nebo Pythonu. V této práci se budu především zabývat již předdefinovanými funkcemi. Pro přehlednost jsem tyto funkce rozdělila do několika kategorií podle způsobu práce s daty podle doc. Ing. Jana Koláře, CSc. [3].

Komunikace s analytickými funkcemi se provádí pomocí dotazů jazyka SQL, nebo jsou tyto dotazy přímo součástí analytických funkcí.

### 10.1 Funkce pro analýzu dat

#### 10.1.1 Výběrové funkce

Výběrové funkce umožňují vyhledávat objekty podle zadaných atributů ať už z popisných či prostorových dat. Výběrové odpovědi jsou zpravidla ve formě tabulky, ale mohou mít i mapový charakter, záleží na struktuře dotazu.

#### 10.1.2 Výpočtové funkce

Umožňují výpočetní operace s popisnými atributy nebo prostorovými daty. Nejčastěji používanými funkcemi jsou SUM, MIN, MAX a AVG. V SQL se tyto funkce nazývají také agregační.

SUM – výsledkem je součet vybraných hodnot

MIN, MAX – výsledkem je minimum nebo maximum z vybraných hodnot

AVG – výsledkem je průměr z vybraných hodnot

### 10.1.3 Analytické funkce

#### 10.1.3.1 Měřicí funkce

##### 10.1.3.1.1 Generalizace

Generalizace znamená zjednodušení a zevšeobecnění geometrie v mapách, také se takto může nazývat zmenšení počtu tříd sloučením několika tříd do jedné (např. sloučení z 5 intervalů počtu obyvatel v okrese na 3 intervaly).

Ke zjednodušení geometrie slouží funkce `simplify(geometry, tolerance)`, kde parametr `tolerance` je maximální povolená kolmá vzdálenost mezi generalizovaným bodem a generalizující linií.

##### 10.1.3.1.2 Měření

Operace měření jsou základními funkcemi, využívající prostorové informace. Pomocí nich je možné měřit obvod `perimeter(geometry)`, výměru `area(geometry)`, v některých případech je výměra objektu již součástí atributové tabulky, délku linií `length(geometry)` a dále využívat funkci vzdálenost od objektu `distance(geometry, geometry)`. Měřicí funkce se zpravidla využívají spolu s následným výběrem, viz příklad 11-7 Area.

#### 10.1.3.2 Funkce překrytí

Funkce překrytí pracují se dvěma nebo více datovými vrstvami (tabulkami), jejichž společnou vlastností je, že pokrývají stejné území. Jejich výsledkem je nový datový soubor s novými atributy a objekty.

Nejčastěji používanými funkcemi překrytí je ořezání a průnik:

Ořezání: Jedna vrstva ořezává svým tvarem vrstvu druhou, výsledná vrstva získává atributy druhé vrstvy.

Průnik `intersection(geometry, geometry)`: Výsledkem je překryv vstupních vrstev s atributy obou vstupních vrstev.

#### 10.1.3.3 Funkce v okolí

Jedná se o zjišťování vlastností v oblasti kolem výchozího objektu. Hledanou hodnotou je atribut, který se buď nachází v databázi, nebo ho funkce vypočítá. K užití funkce v okolí musíme obvykle

zadat minimálně 3 parametry: výchozí objekt, velikost oblasti okolo objektu a definici funkce v okolí.

#### *10.1.3.3.1 Vyhledávací funkce*

Používá se k vyhledání objektu, který splňuje zadanou podmínku. V definici funkce určíme objekt, který se má v okolí výchozího bodu hledat. Toto okolí se též nazývá vyhledávací oblast.

Vyhledávací oblast je často stanovena uměle např. kružnicí, nebo může nabývat i nepravidelných tvarů – např. hranice krajů.

Příkladem uměle stanovené hranice může být vyhledání přírodních památek 50 km od obce.

#### *10.1.3.3.2 Funkce blízkosti*

Funkce blízkosti zjišťuje ve všech sousedních bodech výchozího objektu hodnotu určitého atributu. Podmínka, kterou hodnota atributu musí splnit, je v tomto případě nerovnice, to znamená, že podmínka je splněna, pokud je hodnota atributu menší nebo rovna zadané limitní hranici. Příkladem atributu může být vzdálenost, ale třeba také hluk, průtok korytem řeky apod. Výsledkem funkce blízkosti je vyrovnávací zóna neboli buffer – ten je většinou jen mezikrokem nezbytným pro další analýzy.

V PostGIS vypadá funkce buffer takto: `buffer(geometry, double precision)`



## 11 Praktické příklady prostorových funkcí

K dispozici byla získána data ZABAGED z oblastí v okolí Lovosic, Turnova, Starého Města a Jáchymova ve formátu shapefile – jedná se o jednotlivé vrstvy veškerých objektů nacházejících se v těchto oblastech např. budovy, vrstva lesa, vrstva ulic, vrstva železničních tratí, železničních zastávek apod. Tato data nakonec nebyla použita. Další geografické soubory jsem získala z předmětu K153GIS2. Tato data se týkají celé České republiky – konkrétně šlo o polygonové vrstvy okresů, obcí, kladu mapových listů TM50, lesů, liniové vrstvy železnic a bodové vrstvy obcí a železničních stanic – také ve formátu shapefile.

Data byla převedena pomocí programu shp2pgsql.exe do SQL a poté vložena do databáze.

Zde jsou uvedené praktické příklady funkcí, které jsou popsány v předchozí kapitole.

## 11.1 Výběrové funkce

*Původní data:* vrstva kladů map TM50

*Příkaz:*

**Příklad 11-1** SELECT

```
SELECT area,perimeter,tm_id, tm  
FROM kltm50  
WHERE tm50='M-33-1-B'
```

*Výsledek:* vybraná data listu M-33-1-B

area	perimeter	tm50_id	tm50
319131104	71507.5	1	"M-33-1B"

*Původní data:* vrstva obcí ČR

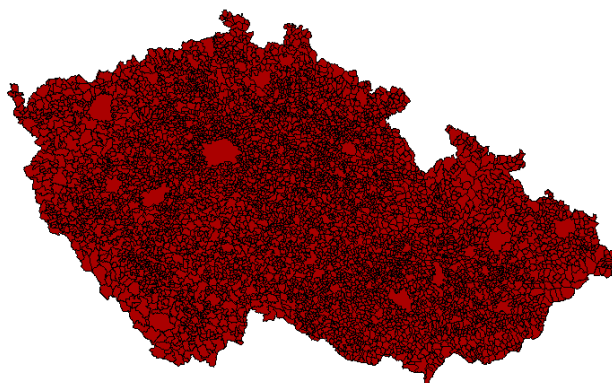
*Příkaz:*

**Příklad 11-2** SELECT

```
SELECT nazev  
FROM obce  
WHERE nazorp='Říčany' AND ob_311202>2000
```

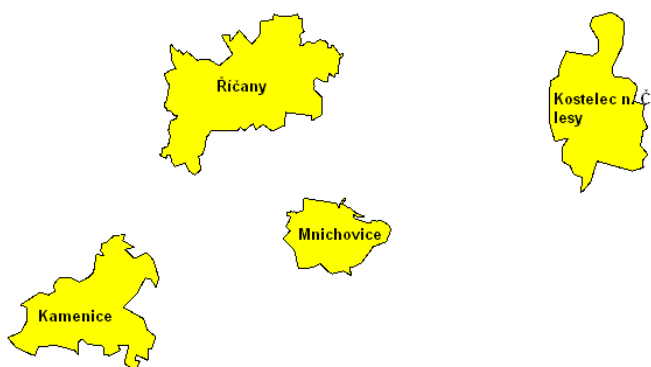
*Výsledek:* výpis obcí, které spadají pod obec s rozšířenou působností Říčany a zároveň mají více než 2000 obyvatel

```
nazev  
-----  
Kostelec nad Černými Lesy  
Kamenice  
Mnichovice  
Říčany
```

**Původní data:***Obrázek 11-1 Vrstva obcí ČR***Příkaz:****Příklad 11-3 SELECT, s grafickým zobrazením**

```
CREATE TABLE orpricany AS  
SELECT gid,nazev, area, the_geom  
FROM obce  
WHERE nazorp='Říčany' AND ob_311202>2000
```

**Výsledek:** výpis obcí, které spadají pod obec s rozšířenou působností Říčany a zároveň mají více než 2000 obyvatel

*Obrázek 11-2 Obce nad 2000 obyvatel, ORP Říčany*

**Původní data:** vrstva obcí ČR

**Příkaz:**

**Příklad 11-4 SELECT**

```
SELECT nazev  
FROM obce  
WHERE area>200000000
```

**Výsledek:** výpis názvů obcí, které mají rozlohu větší než 200 km<sup>2</sup>

```
nazev  
-----  
Libavá  
Brdy  
Boletice  
Praha  
Ostrava  
Hradiště  
Brno
```

## 11.2 Výpočtové funkce

### 11.2.1 Funkce SUM

*Původní data:* vrstva železnic ČR

*Příkaz:*

**Příklad 11-5 SUM()**

```
SELECT sum(length(the_geom))/1000 AS km_zeleznice
FROM zeleznice
```

*Výsledek:* celková délka železnic v ČR v km

```
km_zeleznice
-----
8720.98579854887
```

### 11.2.2 Funkce MIN

*Původní data:* vrstva obcí ČR

*Příkaz:*

**Příklad 11-6 MIN()**

```
SELECT nazev, area AS m2_area
FROM obce
WHERE area=(SELECT MIN(area) FROM obce)
```

*Výsledek:* obec s nejmenší plochou v ČR

```
nazev                m2_area
-----
Karlova Studánka    399151
```

## 11.3 Měřicí funkce

### 11.3.1 Funkce výměra

*Původní data:* vrstva okresů ČR

*Příkaz:*

#### Příklad 11-7 Area(geometry)

```
SELECT okres, (area(okresy.the_geom))/1000000 AS vymera_km2
FROM okresy
WHERE okres='Děčín'
```

*Výsledek:* výměra okresu Děčín

```
okres  vymera_km2
-----
Děčín  905.04083076909
```

### 11.3.2 Funkce obvod

*Původní data:* vrstva okresů ČR

*Příkaz:*

#### Příklad 11-8 Perimeter(geometry)

```
SELECT okres, perimeter(the_geom) AS obvod_km
FROM okresy
WHERE perimeter(the_geom)=(SELECT MAX(perimeter(the_geom))
FROM okresy)
```

*Výsledek:* okres s nejdelší hranicí v ČR

```
okres  obvod_km
-----
Jindřichův Hradec  289.827384974142
```

### 11.3.3 Funkce vzdálenost

**Původní data:** bodová vrstva obcí ČR a bodová vrstva železničních stanic ČR

**Příkaz:**

#### Příklad 11-9 Distance(geometry, geometry)

```
SELECT distance(zel_stan.the_geom, obce_b.the_geom)/1000 AS
vzdalenost_km, zel_stan.nazev
FROM zel_stan, obce_b
WHERE distance(zel_stan.the_geom, obce_b.the_geom) < 10000 AND
obce_b.nazob = 'Chocerady'
```

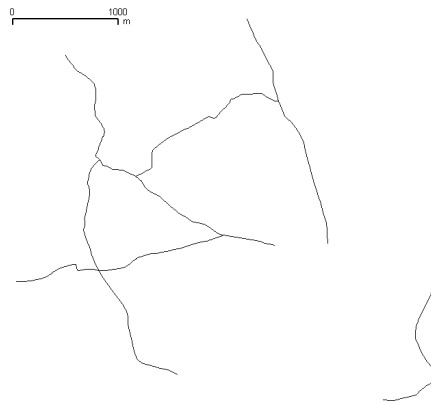
**Výsledek:** železniční stanice do vzdálenosti 10 km od obce Chocerady s vypočtenou vzdáleností

vzdalenost_km	nazev
9.89215548683811	Poříčí n.Sázavou
8.37760295394251	Mrač
7.93593767621579	Mirošovice u Prahy
6.14354280837053	Senohraby
5.8414095955086	Čtyřkoly
7.28987832787052	Pyšely
7.51410015404373	Čerčany
5.92763241528869	Lštění
4.9638615323657	Zlenice
1.94775276861727	Hvězdonice
0.734001160165977	Chocerady
1.59680795847528	Vlkovec
3.2290192644449	Samechov
4.09039015941634	Stříbrná Skalice
5.51424045370258	Plužiny
7.6881873727248	Sázava zast.
7.73437210328229	Sázava-Černé Budy
8.22504808591567	Samopše
9.30591878272237	Ledečko

### 11.3.4 Funkce generalizace

**Původní data:** silnice v obci Třebenice, získané pomocí tohoto příkazu:

```
CREATE TABLE prunik_sil AS
SELECT Intersection(obce.the_geom, silnicedalnice.the_geom)
AS intersection_geom, obce.nazev, obce.gid
FROM obce, silnicedalnice
WHERE intersects(obce.the_geom, silnicedalnice.the_geom ) AND
obce.nazev = 'Třebenice'
```



Obrázek 11-3 Silnice v obci Třebenice

**Příkaz:**

**Příklad 11-10 Simplify(geometry, tolerance)**

```
CREATE TABLE gener AS  
SELECT simplify(prunik_sil.intersection_geom,50), oid FROM  
prunik_sil
```

**Výsledek:** Generalizace silnic v obci Třebenice, červenou barvou je vyznačena generalizovaná vrstva, černě původní data.



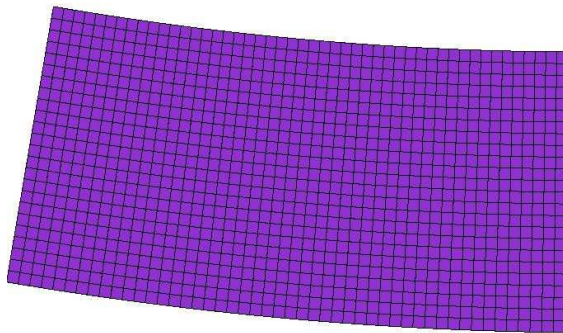
Obrázek 11-4 Generalizace silnic v obci Třebenice



## 11.4 Funkce překrytí

### 11.4.1 Funkce průnik

*Původní data:*



Obrázek 11-5 Klad listů TM50



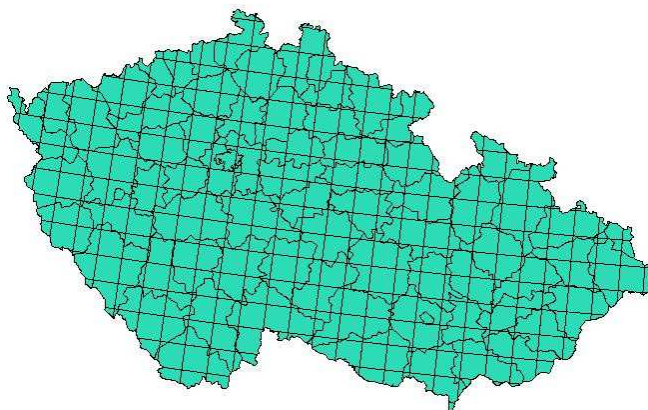
Obrázek 11-6 Vrstva okresů ČR

*Příkaz:*

#### Příklad 11-11 Intersection(geometry, geometry)

```
CREATE TABLE prunik AS  
SELECT Intersection(okresy.the_geom, kltm50.the_geom) AS  
intersection_geom,  
kltm50.*  
FROM okresy, kltm50  
WHERE intersects(okresy.the_geom, kltm50.the_geom )
```

*Výsledek:* omezení kladu listů TM50 na území ČR



Obrázek 11-7 Omezení kladu listů jen na území ČR

## 11.5 Funkce v okolí

### 11.5.1 Vyhledávací funkce

*Původní data:* bodová vrstva obcí ČR a polygonová vrstva okresů ČR

*Příkaz:*

#### Příklad 11-12 Vyhledávací funkce

```
SELECT obce_b.nazob FROM obce_b, okresy
WHERE okresy.okres='Šumperk' AND
contains(okresy.the_geom,obce_b.the_geom)='t'
ORDER BY obce_b.nazob
```

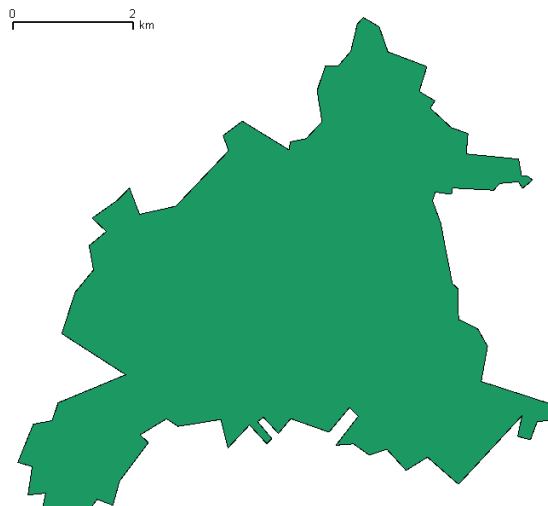
*Výsledek:* obce v okrese Šumperk, seřazené abecedně podle názvu

```
nazob
-----
Bludov
Bohdíkov
Bohuslavice
Bohutín
Branná
Bratrušov
Brníčko
Bušín
Dlouhomilov
Dolní Studénky
Drozdov
Dubicko
Hanušovice
Horní Studénky
Hoštejn
Hraběšice
Hrabišín
Hrabová
Hynčina
Chromeč
Jakubovice
Janoušov
Jedlí
Jestřebí
Jindřichov
Kamenná
Klopina
...
```

### 11.5.2 Funkce blízkosti

**Původní data:** hranice obce Kolín, získané pomocí příkazu:

```
CREATE TABLE kolín AS  
SELECT obce.gid, obce.the_geom  
FROM obce  
WHERE nazev='Kolín'
```



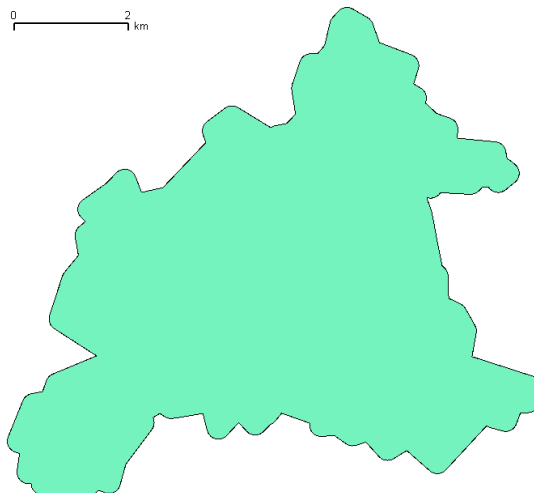
Obrázek 11-8 Hranice obce Kolín

**Příkaz:**

#### Příklad 11-13 Buffer(geometry, double precision)

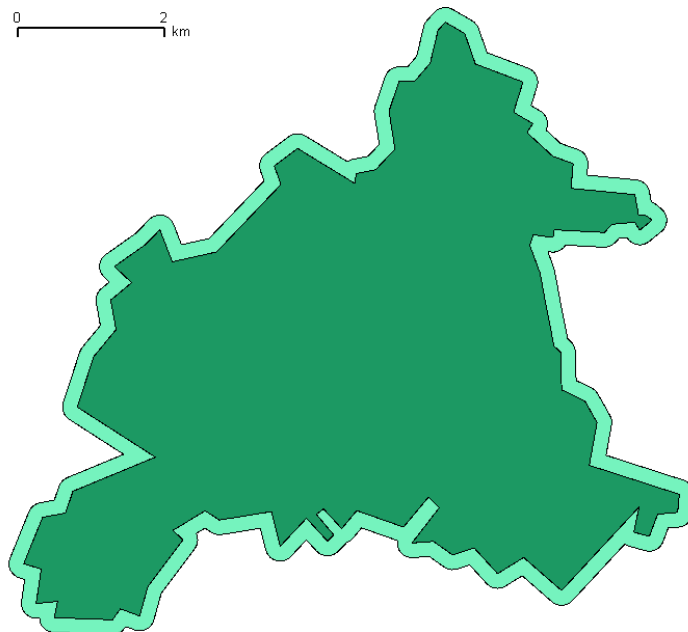
```
CREATE TABLE buffer AS  
SELECT buffer(obce.the_geom, 200), obce.gid, obce.the_geom  
FROM obce  
WHERE nazev='Kolín'
```

**Výsledek:** obalová zóna ve vzdálenosti 200 m okolo obce Kolín



Obrázek 11-9 Buffer 200 m

Pro názornost můžeme obě vrstvy položit na sebe:



*Obrázek 11-10 Buffer 200 m s obrysem obce Kolín*

## 12 Závěr

V úvodu práce jsou představeny možnosti ukládání prostorových dat a databázový systém PostgreSQL a jeho rozšíření PostGIS. Dále byl vysvětlen průběh instalace pro OS Windows a seznámení se základními objekty PostGIS a normou „Simple Features for SQL“. Dále jsou představeny různé prohlížeče prostorových dat, z nichž byl prakticky použit především Quantum GIS. Byl také řešen problém Křovákova zobrazení a teoreticky byly také vysvětleny analytické funkce, které PostGIS obsahuje.

V praktické části byly některé analytické funkce využity na konkrétních příkladech. Jednotlivé ukázky vždy obsahují popis původních dat, SQL příkaz, výsledná data a případně také zobrazení výsledné vrstvy.

Vzhledem k tomu, že tato práce obsahuje nejen teoretický úvod, ale také praktické ukázky možného použití různých analytických funkcí, mohl by být tento dokument vhodný jako učební pomůcka pro výuku předmětu zabývající se tématem prostorových databází, který na naší fakultě zatím chybí.

Další možností rozšíření této práce je vytvoření webových stránek s prezentací této práce, určených taktéž k výuce tohoto předmětu, které by také obsahovaly např. zadání další cvičebních úloh, hodnocení studentů apod.

## 13 Seznam použité literatury

1. **Wikimedia Foundation, Inc.,** PostGIS - Wikipedia, the free encyclopedia. *Wikipedia*. [Online] 18. březen 2008. [Citace: 29. březen 2008.] [http://en.wikipedia.org/wiki/GRASS\\_GIS](http://en.wikipedia.org/wiki/GRASS_GIS).
2. Chapter 4. Using PostGIS. *PostGIS*. [Online] 14. březen 2008. [Citace: 15. březen 2008.] <http://postgis.refractor.net/docs/ch04.html>.
3. **Doc. Ing. Kolář, Jiří Csc.** *Geografické informační systémy 10*. Praha : ČVUT, 2003. ISBN 80-01-02687-6.
4. **PostgreSQL Global Development Group** . PostgreSQL: Documentation: manuals. *PostgreSQL*. [Online] PostgreSQL Global Development Group , 2008. [Citace: 20. duben 2008.] <http://www.postgresql.org/docs/8.2/static/installation.html>.
5. Soubor Gism1.png - GRASSwikiCZ. *GRASSwikiCZ*. [Online] 3. únor 2007. [Citace: 20. duben 2008.] <http://grass.fsv.cvut.cz/wiki/index.php/Soubor:Gism1.png>.
6. **Open Geospatial Consortium, Inc.** Simple Feature Access. *OGC Website*. [Online] 2008. [Citace: 30. duben 2008.] [http://portal.opengeospatial.org/files/?artifact\\_id=829](http://portal.opengeospatial.org/files/?artifact_id=829).
7. **Strnad, Michal.** *Topologické operace v geografických informačních systémech*. 2005.
8. **Bořík, Milan a Vojtěch, Honzík.** GIS Ostrava 2005. *Institut Geoinformatiky*. [Online] 2008. [Citace: 20. 3 2008.] [http://gis.vsb.cz/GIS\\_Ostrava/GIS\\_Ova\\_2005/Sbornik/prez/Borik.ppt](http://gis.vsb.cz/GIS_Ostrava/GIS_Ova_2005/Sbornik/prez/Borik.ppt).
9. **Kamler, Jaromír.** *Analýza možností SŘBD PostgreSQL a nadstavby pro vytvoření datového skladu*. Ostrava : autor neznámý, 2006.
10. **ESRI.** Esri GIS Mapping Software. *Desktop GIS*. [Online] 2008. [Citace: 21. duben 2008.] [http://www.esri.com/software/arcgis/graphics/social\\_distance\\_1g.jpg](http://www.esri.com/software/arcgis/graphics/social_distance_1g.jpg).
11. **Zubíček, Jan.** *GIS JAKO SYSTÉM PRO KOMUNIKACI SE ZÁKAZNÍKY*. Praha : 2007.