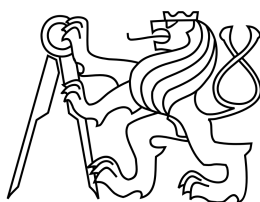


ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE



**FAKULTA STAVEBNÍ
OBOR GEOINFORMATIKA
KATEDRA MAPOVÁNÍ A KARTOGRAFIE
DOKUMENTACE
ÚVOD DO ZPRACOVÁNÍ PROSTOROVÝCH DAT
SEMESTRÁLNÍ PROJEKT**

SKUPINA c13
Jana Antlová, Michaela Šípková
Letní semestr 2012/2013

28.5.2013

Obsah

1 Úvod	2
1.1 Zadání	2
1.2 Použitý software	2
2 Zdroje dat	3
2.1 OpenStreetMap	3
2.2 Schéma OSM	3
3 Tvorba tématických vrstev a jejich validace	4
3.1 Bodová vrstva	4
3.1.1 Kino	4
3.1.2 Bankomaty	5
3.2 Liniová vrstva	6
3.2.1 Potrubí	6
3.2.2 Silnice	7
3.3 Polygonová vrstva	9
3.3.1 Nákupní centra	9
3.3.2 Obce	10
3.3.3 Kraje	10
3.4 Validace	10
4 Použité funkce	11
5 Dotazy	12
5.1 Atributové dotazy	12
5.2 Prostorové dotazy	13
6 Závěr	19
6.1 Zdroje	19

1 Úvod

Tato dokumentace vznikla v rámci předmětu Úvod do zpracování prostorových dat (153UZPD). Semestrální projekt je zaměřen na geoprostorová data a jejich následné zpracování prostřednictvím databáze PostGIS. V dokumentaci je popsán postup tvorby vrstev, transformace dat, případná úprava dat, přehled použitých funkcí a na závěr jsou uvedeny atributové a prostorové dotazy týkající se těchto vrstev.

Během tvorby projektu byly využity znalosti z předmětů Databázové systémy, GIS1 a GIS2.

1.1 Zadání

Navrhněte a vytvořte tématické vrstvy (např. vodní toky, vodní plochy, lesy, silnice, železnice a pod.) na základě dat *OpenStreetMap* (viz cvičná databáze *pgis_student* schéma *osm*). Pro tento účel byla na serveru *'geo102'* založena databáze *pgis_uzpd*.

Aplikujte testy datové integrity a odstraňte případné nekonzistence v datech.

Vytvořte tutoriál pro výuku *PostGIS* - tj. sadu atributových a prostorových dotazů nad databází *pgis_uzpd*.

1.2 Použitý software

PgAdmin III 1.16

Quantum GIS Lisboa (1.18.0)

MiKTeX 2.9

TexMaker 4.0.1

2 Zdroje dat

2.1 OpenStreetMap

Projekt vznikl v červenci 2004. Cílem je tvorba svobodných geografických dat a následně jejich vizualizace do podoby topografických dat. Podkladem jsou záznamy z přijímačů GPS, které jsou sbírány dobrovolníky. Zdrojem jsou také data z komerčních firem, například satelitní a letecké snímky, které poskytla společnost Yahoo.

Další informace o tomto projektu naleznete na stránkách <http://cs.wikipedia.org/wiki/OpenStreetMap>.

2.2 Schéma OSM

Podkladem pro tvorbu vrstev jsou data z projektu OpenStreetMap, získána z databáze pgis_uzpd umístěné na serveru <http://geo102.fsv.cvut.cz/> ze schématu osm a také data ze schématu gis1.

Schéma osm obsahuje data České republiky z projektu OpenStreetMap. V tomto schématu nalezneme celkem sedm vrstev:

```
czech_line  
czech_nodes  
czech_point  
czech_polygon  
czech_rels  
czech_roads  
czech_ways
```

Schéma gis1 obsahuje data ze cvičení k předmětu GIS1 (153GIS1). Najdeme zde vrstvy:

```
kltm50  
lesy  
obce  
obce_b  
obce_s  
zeleznice  
zeleznice_stanice
```

3 Tvorba tématických vrstev a jejich validace

Nejprve bylo založeno schéma pro naši skupinu, schéma 'c13'.

Po prohlédnutí schématu `osm` jsme se rozhodly vytvořit 7 tématických vrstev. Snažily jsme se vybírat tak, aby se vrstvy neopakovaly v rámci předešlých projektů. Všechny nově vytvořené tématické vrstvy jsou v jednotném souřadnicovém systému EPSG:900913 (Spherical Mercator) a obsahují sloupec s geometrií `geom`.

Na začátku byla nastavena cesta ke schématům.

```
SET search_path TO c13, gis1, osm, public;
```

Pro každou vrstvu byl nastaven primární klíč a definován prostorový index nad sloupcem `geom` sloužící ke zrychlení prostorových dotazů.

3.1 Bodová vrstva

3.1.1 Kino

Data pro tvorbu vrstvy 'kino' byly bodové i polygonové. Postup spočíval v tom, že jsme nejprve vytvořily tabulku 'kino' (bodová vrstva). Poté jsme vytvořily polygonovou vrstvu. Pokud se stalo, že by některé kino z bodové vrstvy leželo uvnitř polygonu, bylo toto kino z bodové vrstvy odstraněno. Dále jsme u polygonů vytvořily centroidy a přidaly je k ostatním bodovým datům. Vizualizací v QGISu jsme narazily na kina, která neležela v ČR (došlo k odstranění z tabulky). Ještě byly z tabulky odstraněny názvy, kterým odpovídala hodnota 'NULL'. Na závěr byly smazány pomocné tabulky.

```
CREATE TABLE kino AS
SELECT osm_id, geom, amenity, name AS nazev
FROM czech_point WHERE amenity='cinema';
```

```
DELETE FROM kino WHERE nazev IS NULL;
```

```
CREATE TABLE kino_pol AS
SELECT osm_id, geom, amenity, name AS nazev
FROM czech_polygon WHERE amenity='cinema';
```

```
DELETE FROM kino_pol WHERE nazev IS NULL;
```

Odstranění duplicit.

```
DELETE FROM kino WHERE
kino.geom IN
(SELECT kino.geom FROM kino
JOIN kino_pol
```

```
ON ST.Within(kino.geom, kino_pol.geom));
```

Vytvoření pomocné tabulky a naplnění bodové tabulky.

```
CREATE TABLE centr AS
SELECT osm_id, ST_Centroid(kino_pol.geom) AS geom,
amenity, nazev FROM kino_pol;
INSERT INTO kino(osm_id, geom, amenity, nazev)
SELECT osm_id, geom, amenity, nazev FROM centr;
```

Odstranění dat mimo ČR.

```
DELETE FROM kino
WHERE kino.geom NOT IN
(SELECT kino.geom FROM kino
JOIN kraje ON ST_Within(kino.geom, kraje.geom));
```

Nastavení primárního klíče.

```
ALTER TABLE kino ADD PRIMARY KEY (osm_id);
```

Nastavení prostorového indexu.

```
CREATE INDEX kino_geom ON kino USING GIST (geom);
```

Odstranění pomocných tabulek.

```
DROP TABLE kino_pol; DROP TABLE centr;
```

3.1.2 Bankomaty

Vrstva 'bankomaty' byla vytvořena z tabulky 'czech_point', kde atribut občanská vybavenost 'amenity'='atm'. Tabulka obsahuje sloupec 'osm_id', 'operator', 'geom' a 'gid'. V atributu 'operator' jsou uloženy názvy společností (příklad Česká spořitelna). U některých záznamů tento údaj chybí. I tak byly v tabulce ponechány. Zobrazením vrstvy v QGISu jsme zjistily, že se několik bankomatů nachází mimo ČR, proto byly z tabulky odstraněny.

Vytvoření tabulky.

```
CREATE TABLE bankomaty AS
SELECT osm_id, operator, geom
FROM czech_point
WHERE amenity='atm';
```

Poté byl nastaven primární klíč.

```
ALTER TABLE bankomaty ADD PRIMARY KEY (osm_id);
```

Vytvoření prostorového indexu.

```
CREATE INDEX bankomaty_geom ON bankomaty USING GIST(geom);
```

Odstranění zahraničních bankomatů.

```
DELETE FROM bankomaty
WHERE bankomaty.geom NOT IN
(SELECT bankomaty.geom FROM bankomaty
JOIN kraje ON ST_Within(bankomaty.geom, kraje.geom));
```

3.2 Liniová vrstva

3.2.1 Potrubí

Nejprve jsme si jako liniovou vrstvu vybraly "potrubí" v ČR. Po zobrazení vrstvy v QGISu jsme zjistily značnou neúplnost dat. Proto jsme tuto vrstvu nezahrnuly do dotazování.

```
CREATE TABLE potrubí AS
SELECT osm_id, name, operator, layer, geom FROM czech_line
WHERE man_made='pipeline';
```

```
ALTER TABLE potrubí
ADD PRIMARY KEY (osm_id);
```

```
CREATE INDEX potrubí_geom ON potrubí USING GIST(geom);
```



Obrázek 1: Ukázka vrstvy

3.2.2 Silnice

Naší jedinou liniovou vrstvou se stala vrstva silnic. Z tagu 'highway' jsme vybraly: dálnice a nájezdy, rychlostní komunikace a nájezdy a na závěr silnice I.,II., III. třídy a k tomu jejich nájezdy.

Některé silnice vedly do okolních států. Pro odstranění byla vytvořena pomocná polygonová vrstva 'hranice'. A pomocí ní byly přesahující úseky silnic vymazány.

```
CREATE TABLE silnice AS
SELECT osm_id, name AS nazev, highway AS popis, geom FROM osm.czech_line
WHERE highway IN ('motorway', 'motorway_link', 'trunk', 'trunk_link', 'primary',
'primary_link', 'secondary', 'secondary_link', 'tertiary', 'tertiary_link');
```

Primární klíč a prostorový index

```
ALTER TABLE silnice
ADD PRIMARY KEY (osm_id);
```

```
CREATE INDEX silnice_geom ON silnice
USING GIST (geom);
```

```
CREATE TABLE hranice AS
SELECT ST_Union(geom) AS geom
FROM c13.obce;
```

Ořezání silnic, které přesahovaly mimo ČR

```
DELETE FROM silnice
WHERE silnice.geom NOT IN
(SELECT silnice.geom FROM silnice
JOIN hranice ON ST_Within(silnice.geom, hranice.geom));
```

Protože jsme zjistily, že u spousty silnic chybí názvy, rozhodly jsme se upravit alespoň část, a to dálnice. Nejprve jsme vytvořily samostatnou vrstvu 'dalnice'. Poté jsme v QGISu vyhledaly jednotlivé 'id' a těmto 'id' jsme přiřadily příslušný název. V ČR se nachází dálnice: D1, D2, D3, D5, D8 a D11. Na závěr jsme vytvořily vrstvu 'dalniceu', ve které najdeme již sjednocené úseky dálnic.

Použité příkazy:

Vytvoření vrstvy 'dalnice'

```
CREATE TABLE dalnice AS
SELECT osm_id, nazev, popis, geom FROM silnice
WHERE popis = 'motorway';
```

Přidání názvů do vrstvy dálnic (ukázka u D3). Názvy dálnic jsme přidaly i do vrstvy 'silnice'.

```
UPDATE dalnice SET nazev= 'D3'
where osm_id IN
```



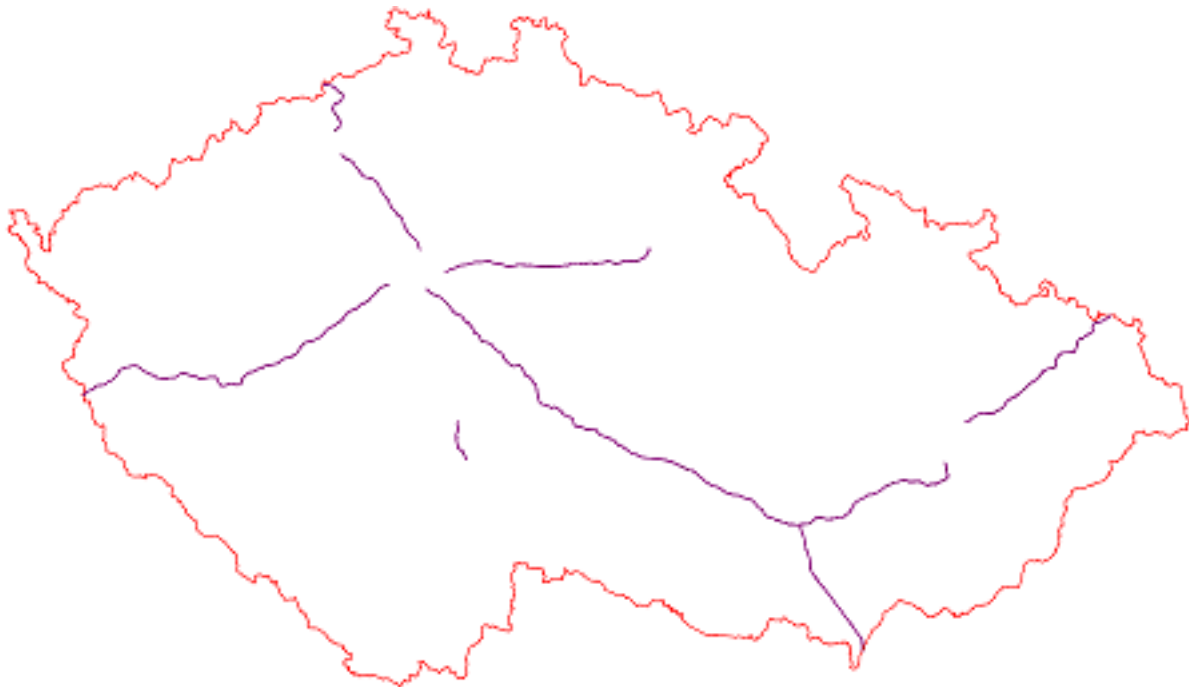
```
( '51526882', '39379554', '102413204', '102413205', '102413203', '83891729', '102413206',  
'39379553', '39379552', '102413207', '51526934', '51526933', '51526926', '51448854',  
'115670155', '51526925', '51448853', '51448749', '83891731', '83891730', '83891732',  
'171049579', '39379549', '171049583', '39379548', '39379547', '39379546', '58920105',  
'171049580', '58920111', '58920116', '58920122', '58933305', '171047170', '32598181',  
'119909466', '171047176');
```

Sjednocení dálnic

```
CREATE TABLE dalniceu AS  
SELECT nazev, ST_Union(geom) AS geom  
FROM dalnice  
GROUP BY nazev;
```

Nastavení primárních klíčů a prostorových indexů na vrstvy.

```
ALTER TABLE dalnice  
ADD COLUMN gid serial  
ADD PRIMARY KEY (gid);  
ALTER TABLE dalniceu  
ADD COLUMN gid serial  
ADD PRIMARY KEY (gid);  
CREATE INDEX dalnice_geom ON dalnice  
USING GIST (geom);  
CREATE INDEX dalniceu_geom ON dalniceu  
USING GIST (geom);
```



Obrázek 2: Ukázka vrstvy 'dalnice'

3.3 Polygonová vrstva

3.3.1 Nákupní centra

Vrstva 'nakupni_centra' byla vytvořena z polygonové tabulky 'czech_polygon'. Atribut jsme se snažily vybrat tak, abychom vytvořily síť větších nákupních center u nás. Podmínkou pro výběr tedy bylo, že atribut 'shop' se rovná 'department_store', 'supermarket' nebo 'mall'.

```
CREATE TABLE nakupni_centra AS
SELECT osm_id, geom, way_area, shop AS druh, name AS nazev
FROM czech_polygon
WHERE shop IN ('mall', 'supermarket', 'department_store');
```

Přejmenování sloupce way_area na plocha.

```
ALTER TABLE nakupni_centra RENAME COLUMN way_area TO plocha;
```

Přidání primárního klíče.

```
ALTER TABLE nakupni_centra
ADD PRIMARY KEY (osm_id);
```

Přidání prostorového indexu.

```
CREATE INDEX nc_geom ON nakupni_centra
USING GIST(geom);
```

Ty objekty, které neměly přiřazený název, jsme kvůli dotazování vyřadily z tabulky.

```
DELETE FROM nakupni_centra WHERE nazev IS NULL;
```

Nakonec jsme ještě odstranily centra, která ležela mimo ČR.

```
DELETE FROM nakupni_centra
WHERE nakupni_centra.geom NOT IN
(SELECT nakupni_centra.geom FROM nakupni_centra
JOIN kraje ON ST_Within(nakupni_centra.geom, kraje.geom));
```

3.3.2 Obce

Tabulka obcí byla převzata ze schématu `gis1`. Tyto data jsou v souřadnicovém systému `S-JTSK` 'Krovak East North', proto je nutné je transformovat. Příkaz pro transformaci byl převzat ze stránek projektu. Pro snadnější orientaci jsme některé sloupce přejmenovali.

3.3.3 Kraje

Na závěr jsme ještě vytvořily polygonovou vrstvu 'kraje', která byla odvozena z vrstvy 'obce'. Tato vrstva nám posloužila při dotazování.

Všechny příkazy jsou uvedené v SQL dávce, která je součástí projektu.

3.4 Validace

Kontrola validace u polygonové vrstvy 'nakupni_centra' proběhla pomocí funkce `ST_IsValid`. Výsledkem je prázdná množina, tedy data jsou validní.

U liniové vrstvy 'silnice' byla kontrola opět provedena pomocí funkce `ST_IsValid`. Ještě jsme u vrstvy 'dalnice' použili funkci `ST_IsSimple`, která kontroluje 'self-intersection'. Výsledek dopadl bez chyb.

```
SELECT osm_id
FROM nakupni_centra
WHERE NOT ST_IsValid(geom);
```

```
SELECT osm_id
FROM silnice
WHERE NOT ST_IsValid(geom);
```

```
SELECT osm_id FROM dalnice
WHERE NOT ST_IsSimple(geom);
```

4 Použité funkce

V této části uvedeme funkce, které jsme použily během dotazování při tvorbě tutoriálu. Jako nápo-
věda nám sloužil manuál na internetových stránkách <http://postgis.net/docs/>.

ST_Within

- pokud je geometrie A celá obsažena v geometrii B, vrátí hodnotu TRUE.

ST_Intersects

- pokud se geometrie A, B překrývají nebo se dotýkají, funkce vrátí hodnotu TRUE.

ST_Intersection

- funkce vrátí geometrický objekt, který je výsledkem průniku geometrií A, B.

ST Area

- vrátí plochu (hodnota typu `float`) zadané geometrie (polygon, multipolygon), pokud je vstupem `geometry`, vrátí plochu v jednotkách SRID.

ST_Distance

- funkce vrací nejkratší kartézskou vzdálenost ve 2D mezi geometrickými objekty typu `float`.

ST_Centroid

- funkce vrátí geometrii středu zadaného geometrického objektu.

ST_Buffer

- funkce vrátí geometrii, kterou reprezentují všechny objekty obsažené v obalové zóně, velikost obalové zóny se určuje pomocí vzdálenosti (poloměru). Vstupem může být `multi/points`, `multi/linestrings`, `multi/polygons`.

ST_Y, ST_X

- funkce vrátí souřadnici Y/X, vstupem musí být bod, výsledkem je hodnota typu `float`.

ST_Dwithin

- funkce vrací hodnotu TRUE pokud leží geometrie A,B do zadané vzdálenosti.

ST_Length

- funkce vrací 2D vzdálenost z geometrie typu `multi/linestring`.

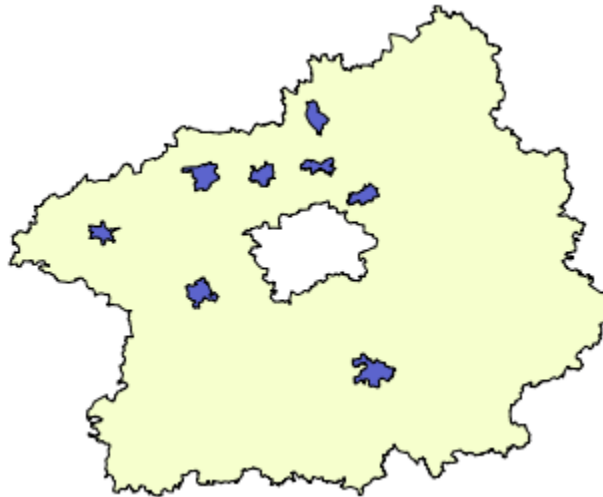
5 Dotazy

5.1 Atributové dotazy

1. Které obce Středočeského kraje mají počet obyvatel mezi 15000 - 20000?

```
SELECT nazev, obyvv02, kraj FROM obce
WHERE obyvv02 > 14999 AND obyvv02 < 20001 AND kraj='Stredocesky'
ORDER BY obyvv02 ASC;
```

```
"Slaný";15165;"Stredocesky"
"Brandýs nad Labem-Stará Boleslav";15287;"Stredocesky"
"Benešov";16277;"Stredocesky"
"Neratovice";16363;"Stredocesky"
"Rakovník";16536;"Stredocesky"
"Kralupy nad Vltavou";17466;"Stredocesky"
"Beroun";17481;"Stredocesky"
"Mělník";19077;"Stredocesky"
```



Obrázek 3: Obce Středočeského kraje s počtem obyvatel 15000 - 20000

2. Kolik je v ČR obchodních domů Prior?

```
SELECT count(nazev) FROM nakupni_centra
WHERE nazev like '%rior%';
```

'4'

3. Jaká je průměrná rozloha (v m²) jednoho nákupního centra v ČR?

```
SELECT ROUND(AVG(plocha)) FROM nakupni_centra;
```

'11716'

4. Najděte největší obchodní centrum v ČR?

```
SELECT osm_id, nazev, plocha, druh  
FROM nakupni_centra AS nc  
ORDER BY plocha DESC limit 1;
```

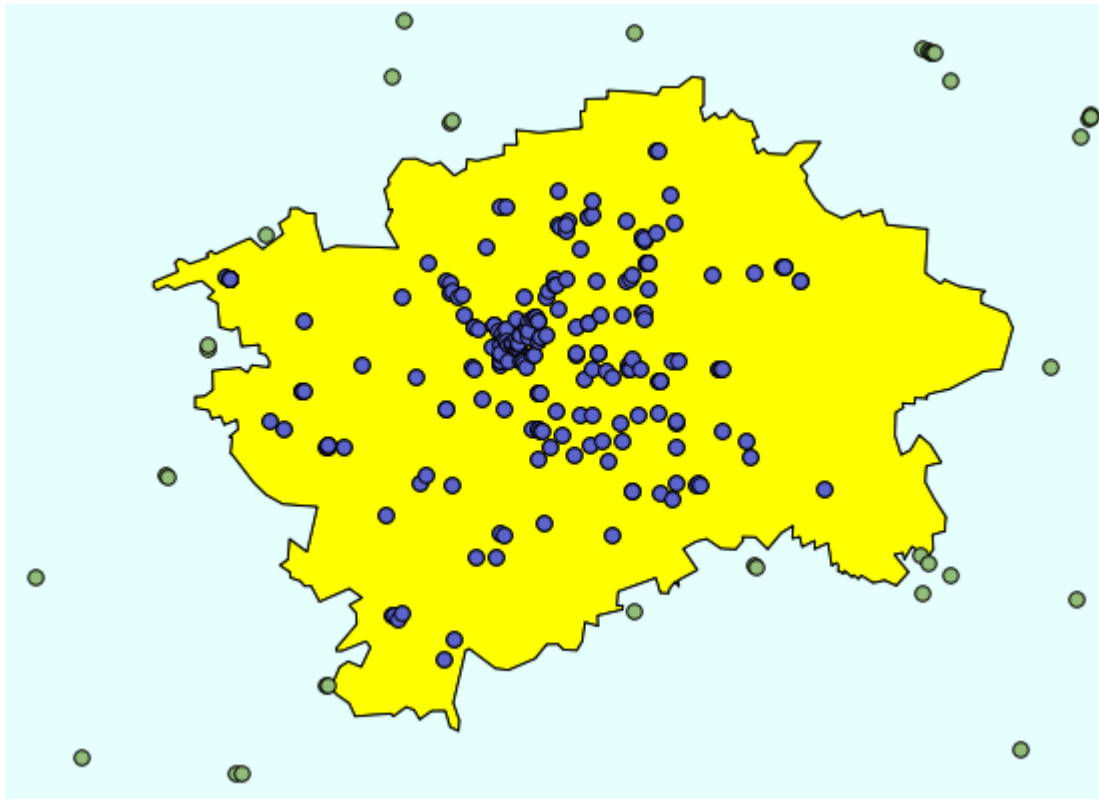
22822822;"SAPA TRADE CENTER";703349;"mall"

5.2 Prostorové dotazy

1. Kolik bankomatů je na území Prahy?

```
SELECT COUNT(*) FROM bankomaty AS b  
JOIN (SELECT geom FROM obce  
WHERE nazev_eng = 'Praha') AS p  
ON ST.Within(b.geom, p.geom);
```

'231'



Obrázek 4: Kolik bankomatů je na území Prahy

2. Jaký kraj má nejvíce obchodních center?

```
SELECT count(nc.nazev), kraj FROM kraje
JOIN nakupni_centra AS nc ON
ST_Intersects(nc.geom, kraje.geom)
GROUP BY kraj ORDER BY count(nazev) DESC
limit 1;
```

106;"Jihomoravsky"

3. Kolik procent území zaujímají nákupní centra?

```
SELECT ROUND(CAST((nc_plocha/plocha)*100 AS numeric),4)
FROM
(
SELECT SUM(ST_Area(ST_Intersection(nc.geom, obce.geom))) AS nc_plocha,
SUM(ST_Area(obce.geom)) AS plocha
FROM obce
JOIN nakupni_centra AS nc
ON ST_Intersects(nc.geom, obce.geom)
) AS nc_cr;
```

"0.046 promile"

4. Uvedte 3 největší nákupní centra a napište v jakém městě se nacházejí.

```
SELECT druh, nc.nazev, o.nazev, plocha
FROM nakupni_centra AS nc
JOIN obce AS o
ON ST_Within(nc.geom, o.geom)
ORDER BY plocha DESC limit 3;
```

"mall";"SAPA TRADE CENTER";"Praha";703349

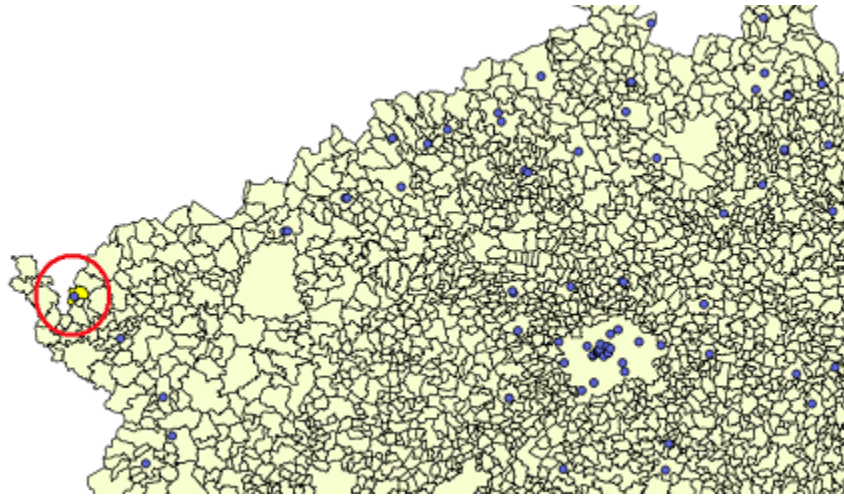
"mall";"Galleria Moda";"Tuchoměřice";125533

"mall";"Obchodní centrum Nisa";"Liberec";111161

5. Jaké je nejzápadnější kino v ČR a v jakém městě se nachází, uveďte jeho souřadnice.

```
SELECT kino.nazev, ROUND(ST_Y(kino.geom)) AS y, ROUND(ST_X(kino.geom)) AS x,
obce.nazev FROM kino
JOIN obce ON ST_Intersects(kino.geom,obce.geom)
ORDER BY x ASC limit 1;
```

"Kulturní dům";6484571;1374484;"Plesná"



Obrázek 5: Nejzápadnější kino ČR

6. Kolik kin je přímo v nákupních centrech?

```
SELECT count(nc.nazev) FROM nakupni_centra AS nc
JOIN kino AS k
ON ST_Intersects(nc.geom, k.geom);
```

"12"

**7. Jak daleko od nejmenší obce v kraji Vysočina jsou nejbližší tři nákupní centra?
Vypište název centra, název obce a vzdálenost.**

```
SELECT obce.nazev, nc.nazev, ROUND(ST_Distance(obce.geom,nc.geom)/1e3)
AS delka_km
FROM obce, nakupni_centra AS nc
WHERE obce.nazev= (SELECT nazev FROM obce
WHERE kraj = 'Vysocina' ORDER BY vymera LIMIT 1)
ORDER BY delka_km
LIMIT 3 ;
```

"Nimpšov";"Billa";8

"Nimpšov";"Plus Diskont Moravské Budějovice";8

"Nimpšov";"Lidl";34

8. Kolik bankomatů je do vzdálenosti 7 km od obce Sedlice?

```
SELECT count(b.operator)
FROM obce AS o
JOIN bankomaty AS b
ON ST_Dwithin(o.geom, b.geom, 7000)
WHERE o.nazev = 'Sedlice';
```


'1'

9. V kterých obcích najdeme kino Cinestar? Napiš název obce a kraj.

```
SELECT DISTINCT(obce.nazev), obce.kraj FROM obce
JOIN kino AS k ON ST_Intersects(k.geom,obce.geom)
WHERE k.nazev like '%ine_tar%';
```

"České Budějovice";"Jihocesky"
"Hradec Králové";"Kralovehradecky"
"Olomouc";"Olomoucky"
"Pardubice";"Pardubicky"
"Ostrava";"Moravskoslezsky"
"Liberec";"Liberecky"
"Plzeň";"Plzensky"
"Jihlava";"Vysocina"
"Praha";"hl.m.Praha"

10. Kde se nachází kino CRYSTAL? Uvedte souřadnice na celé metry.

```
SELECT obce.nazev, kraj, ROUND(ST_X(k.geom)), ROUND(ST_Y(k.geom))
FROM kino AS k
JOIN obce
ON ST_Intersects(obce.geom, k.geom)
WHERE k.nazev = 'Crystal';
```

"Česká Lípa";"Liberecky";1617929;6565740

11. Určete souřadnice nákupního centra Plzeň Plaza.

```
SELECT ROUND(ST_X(ST_Centroid(nc.geom))), ROUND(ST_Y(ST_Centroid(nc.geom))),
nazev
FROM nakupni_centra AS nc
WHERE nazev= 'Plzeň Plaza';
```

1488280;6403015;"Plzeň Plaza"

12. Kolik km silnic II. třídy je ve Zlíně?

```
SELECT ROUND(SUM(ST_Length(s.geom))/1e3)
FROM silnice AS s
JOIN obce
ON ST_Within(s.geom,obce.geom)
WHERE obce.nazev_eng='Zlin'
AND s.popis = 'secondary';
```

'18'

13. Kolik bankomatů je v okolí Vítězného náměstí v Praze (do 250m)?

```
SELECT count((b.operator))
FROM bankomaty AS b
JOIN silnice AS s
ON ST_Dwithin(b.geom, s.geom, 250)
WHERE s.nazev= 'Vítězné náměstí';
```

'2'

14. Které kino leží nejbližší silnici? Napište i vzdálenost.

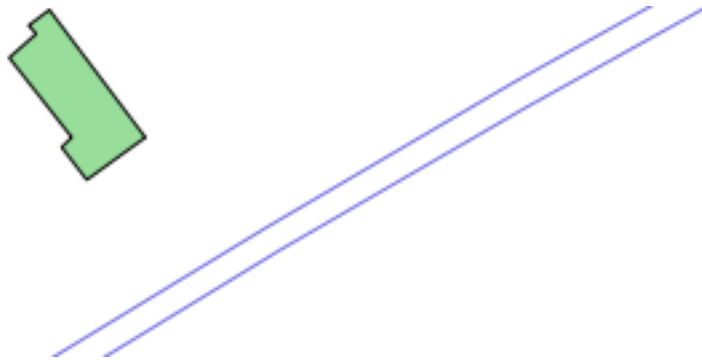
```
SELECT k.nazev, ROUND(ST_Distance(k.geom,s.geom))
FROM kino AS k
JOIN silnice AS s
ON k.geom && s.geom
ORDER BY ST_Distance(k.geom, s.geom)
LIMIT 1;
```

"Kino Krupka";16

15. Která nákupní centra leží do 100m od dálnice D5?

```
SELECT nc.nazev
FROM nakupni_centra AS nc
JOIN silnice AS s
ON ST_Dwithin(nc.geom, s.geom, 100)
WHERE popis='motorway' AND s.nazev = 'D5';
```

"Lidl"



Obrázek 6: Lidl u dálnice D5.

16. Které nákupní centrum leží nejbližší hranicím České republiky?

Nejprve si vytvoříme pomocnou vrstvu 'hranice2'.

```
DROP TABLE hranice2;
```

```
CREATE TABLE hranice2
AS SELECT ST_Boundary(ST_Union(geom)) AS geom
FROM obce;
```

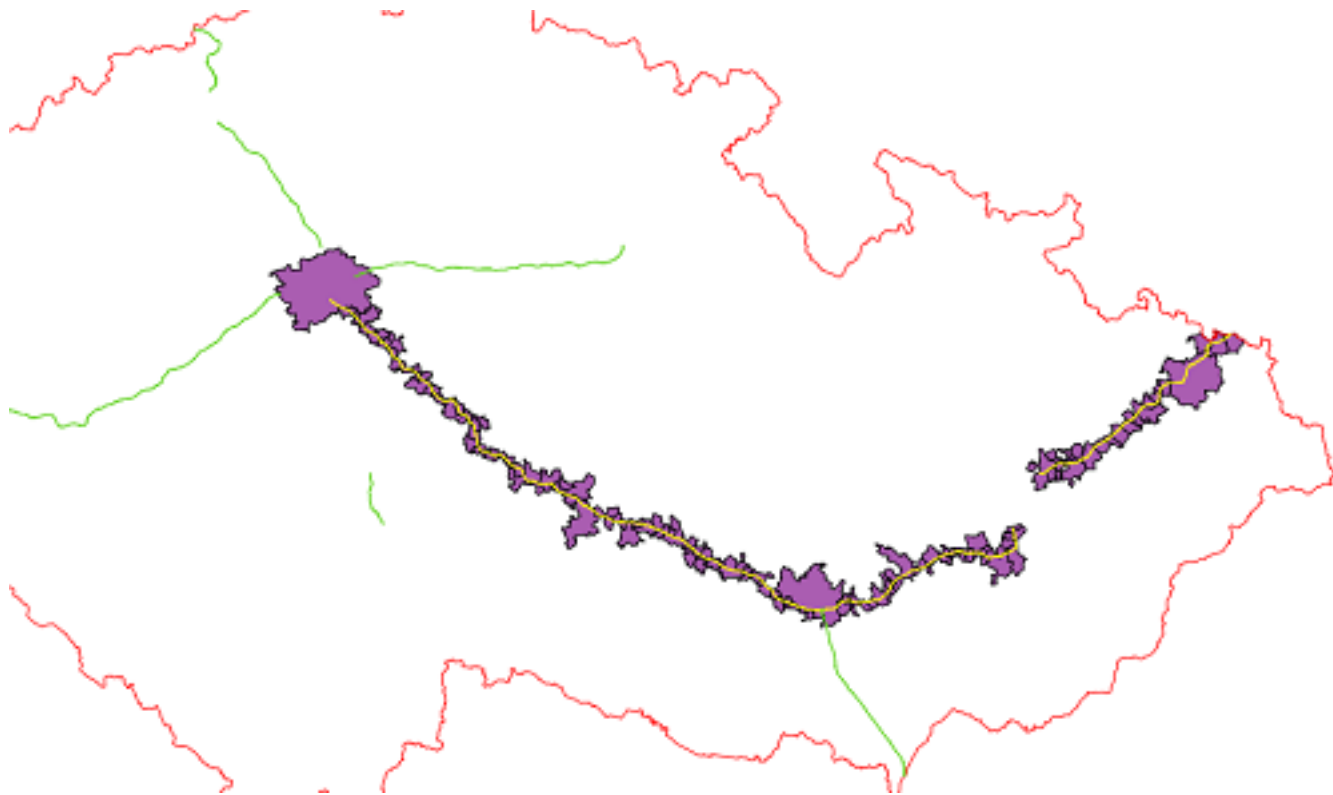
```
SELECT nc.nazev, ROUND(ST_Distance(nc.geom, h.geom))
FROM nakupni_centra AS nc
JOIN hranice2 AS h
ON nc.geom && h.geom
ORDER BY ST_Distance(nc.geom, h.geom)
LIMIT 1;
```

"Albert Hypermarket";415

17. Kterými obcemi prochází dálnice D1?

```
SELECT o.nazev, o.geom
FROM obce AS o
JOIN dalniceu AS d
ON ST_Intersects(d.geom,o.geom)
WHERE d.nazev = 'D1';
```

'138 obcí'



Obrázek 7: Obce, kterými prochází dálnice D1.

6 Závěr

Tato dokumentace slouží k projektu, který vznikl v předmětu Úvod do zpracování prostorových dat. Naší snahou a úkolem bylo vytvořit přehledný a jednoduchý manuál pro práci v PostGIS. Více jsme se zaměřily na tutoriál, kde jsme se snažily využít co nejvíce funkcí a ukázat různé typy dotazů, které se neopakují. Vytvořené vrstvy se nachází ve schématu `c13`, v databázi `pgis_uzpd` na serveru `geo102.fsv.cvut.cz`. Při tvorbě vrstev jsme se potýkaly s řadou problémů, například velké množství dat nemá vyplněné důležité atributy. Některá data navíc nebyla úplná. Také může dojít k duplicitě dat (v našem případě, vrstva kin vytvořená z bodové a polygonové tabulky).

6.1 Zdroje

<http://postgis.net/docs/manual-2.0/>
<http://geo.fsv.cvut.cz/gwiki/153UZPD/>