



FAKULTA STAVEBNÍ

STUDIJNÍ PROGRAM: GEODÉZIE A KARTOGRAFIE

TYP PROGRAMU: MAGISTERSKÝ

STUDIJNÍ OBOR: GEOMATIKA

# ÚVOD DO ZPRACOVÁNÍ PROSTOROVÝCH DAT

## SEMESTRÁLNÍ PROJEKT

**Josef Gruber**

**Pavel Kopecký**

Akademický rok 2015/2016

# Obsah

1	Úvod .....	2
1.1	Zadání.....	2
1.2	Použitý software.....	2
2	Zdroje dat .....	2
2.1	OpenStreetMap.....	2
2.2	Registr územní identifikace, adres a nemovitostí .....	3
3	Datové vrstvy .....	3
3.1	Tvorba vrstvy .....	4
3.1.1	Transformace .....	5
3.1.2	Odstranění zahraničních částí.....	5
4	Použité prostorové funkce.....	5
5	Dotazy .....	6
5.1	Atributové.....	6
5.2	Prostorové.....	8
6	Závěr.....	14
7	Přílohy .....	14
8	Zdroje .....	14

# 1 Úvod

Dokumentace byla vytvořena jako semestrální projekt předmětu Úvod do zpracování prostorových dat na Fakultě stavební Českého vysokého učení technického. Projekt je zaměřen na tvorbu tématických vrstev a použití SQL dotazů nad nimi. Veškeré práce probíhaly v programu QGIS za pomoci rozšíření PostGIS nad databází pgis\_uzpd, ze které byla využita data OpenStreetMap a data Registru územní identifikace, adres a nemovitostí. Výsledná dokumentace poté poskytuje podrobný náhled na veškeré činnosti spojené s tvorbou jednotlivých částí projektu.

## 1.1 Zadání

Návrh a tvorba tématických vrstev (např. vodní toky, vodní plochy, lesy, silnice, železnice a podobně) na základě dat OpenStreetMap a dalších otevřených zdrojů.

Aplikace testů datové integrity a odstranění případných nekonzistencí v datech.

Tvorba tutoriálu - tj. sady atributových a prostorových dotazů nad databází pgis\_uzpd.

## 1.2 Použitý software

QGIS 2.12.2

OpenOffice Writer

Word 2013

# 2 Zdroje dat

Byla využita volně dostupná data OpenStreetMap (dále jen OSM) a data Registru územní identifikace, adres a nemovitostí (dále jen RÚIAN). Z dat OSM byly použity polygonové vrstvy „obce“, „okresy“ a „kraje“. Z dat RÚIAN byla použita liniová vrstva „czech\_roads“, polygonová vrstva „czech\_polygon“ a také bodová vrstva „czech\_point“.

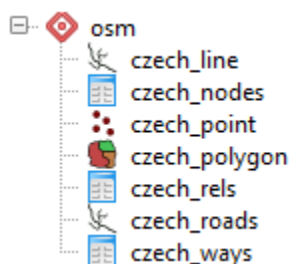
## 2.1 OpenStreetMap

OSM je názvem projektu, který vznikl v roce 2004 a jehož hlavním cílem je tvorba volně dostupných geografických dat a jejich následná vizualizace v podobě topografických map. Pro tvorbu geodat jsou využívány záznamy z GPS přijímačů, družicové snímky nebo další digitalizované mapy.

OpenStreetMap tvoří komunita uživatelů, kteří přidávají a udržují data o silnicích, cestách, kavárnách, železničních stanicích a mnohém dalším po celém světě. Díky

OSM fungují mapová data na tisících webových stránkách, mobilních aplikací a hardwarových zařízeních.

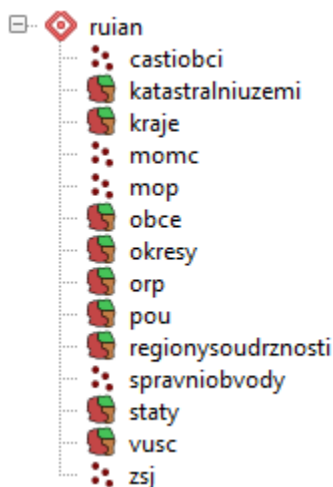
Databáze OSM nám pro Českou republiku poskytla data v 7 různých vrstvách:



## 2.2 Registr územní identifikace, adres a nemovitostí

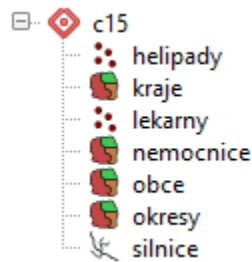
RÚIAN je jedním ze čtyř základních registrů České republiky. Dalšími jsou registr obyvatel (ROB), registr práv a povinností (RPP) a registr osob (ROS). Je spravován Českým úřadem zeměměřickým a katastrálním. Je součástí informačního systému základních registrů. Hlavní funkcí registru RÚIAN je poskytování informačních a servisních služeb pro veřejnost a veřejnou správu.

Databáze RÚIAN nám poskytla data ve 14 různých vrstvách:



## 3 Datové vrstvy

Bylo vytvořeno celkem 7 datových vrstev, které jsou uloženy ve schématu c15 v databázi pgis\_uzpd. Jsou to tyto vrstvy: bodové vrstvy „lekarny“ a „helipady“, liniová vrstva „silnice“ a polygonové vrstvy „obce“, „okresy“, „kraje“ a „nemocnice“.



### 3.1 Tvorba vrstvy

Jak již bylo zmíněno, bylo vytvořeno celkem 7 vrstev. Pro praktickou ukázkou použijeme vrstvu „helipady“ neboli bodovou vrstvu znázorňující přistávací plochy pro vrtulníky.

Nejprve bylo nutné nastavit cestu k jednotlivým databázím a schémátům.

```
SET search_path TO c15, osm, ruian, public;
```

Následně byla vytvořena tabulka s vybranými objekty a parametry.

```
CREATE TABLE helipady AS
SELECT name as nazev_helipadu, aeroway, geom
FROM czech_point
WHERE aeroway='helipad';
```

Byla tedy vytvořena nová vrstva (tabulka) „helipady“, která zobrazuje název přistávací plochy, typ letiště a geometrii. Tato vrstva byla vytvořena z původní vrstvy „czech\_point“.

V dalším kroku byl k jednotlivým vrstvám vytvořen primární klíč „id“ a prostorový index (u transformovaných vrstev (viz. kapitola 3.1.1) byl přiřazen prostorový index až po transformaci).

```
ALTER TABLE helipady ADD COLUMN id serial;
ALTER TABLE helipady ADD PRIMARY KEY (id);
CREATE INDEX helipady_index ON helipady USING GIST(geom);
```

Nakonec byl proveden test datové integrity (validace). K tomu posloužil následující příkaz.

```
SELECT id FROM helipady WHERE not st_isvalid(geom);
```

Kontrola validace nezjistila v datech žádný problém.

Výše uvedené příkazy byly použity u všech vytvořených vrstev.

U polygonových vrstev byly přidány sloupce s výměrou polygonů (m<sup>2</sup>, km<sup>2</sup>), u vrstvy silnice byly přidány sloupce s délkami linií (m, km).

```
ALTER TABLE okresy ADD COLUMN vymera_km2 float;
UPDATE okresy SET vymera_km2 = ST_Area(geom)/1000000;
```

### 3.1.1 Transformace

Byly využity 2 různé zdroje dat s odlišnými referenčními systémy. Databáze RÚIAN poskytuje data v systému S-JTSK (EPSG: 5514). Databáze OSM poskytuje data v systému WGS84 (EPSG: 3857). Aby byl celý projekt v jednotném referenčním systému, byly nově vytvořené vrstvy z dat OSM přetransformovány do systému S-JTSK. Transformace u vrstev „nemocnice“, „helipady“, „lekarny“ a „silnice“ byla provedena níže uvedeným příkazem. Je uveden příklad pro vrstvu „nemocnice“.

```
SELECT AddGeometryColumn('nemocnice', 'g', 5514, 'multipolygon', 2);
UPDATE nemocnice SET g = ST_Transform(geom, 5514);
SELECT DropGeometryColumn('nemocnice', 'geom');
ALTER TABLE nemocnice RENAME COLUMN g TO geom;
```

U vrstev „helipady“ a „lekarny“ byl nastaven typ geometrie 'point', u vrstvy silnice 'linestring'.

### 3.1.2 Odstranění zahraničních částí

U bodových vrstev byly vypuštěny body ležící mimo území České republiky, jelikož tyto body nebyly potřebné pro následné práce. Pro odstranění zahraničních helipadů vypadal příkaz následovně.

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

## 4 Použití prostorové funkce

### ST\_Area

- vrací plochu zadané geometrie

### ST\_Buffer

- vrací geometrii zahrnující všechny body v zadané vzdálenosti od dané geometrie

### ST\_Centroid

- vrací geometrický střed geometrie

### ST\_Disjoint

- vrací hodnotu TRUE, pokud geometrie nesdílí žádnou plochu

### ST\_Distance

- vrací nejkratší kartézskou vzdálenost mezi dvěma geometriemi

### ST\_Dwithin

- vrací hodnotu TRUE, pokud je geometrie v oblasti do zadané vzdálenosti od dané geometrie

## ST\_Intersects

- vrací hodnotu TRUE, pokud se geometrie aspoň částečně překrývají

## ST\_IsValid

- vrací hodnotu TRUE, pokud je geometrie dobře vytvořená

## ST\_Length

- vrací délku zadané geometrie

## ST\_Perimeter

- vrací délku hranice geometrie

## ST\_Touches

- vrací hodnotu TRUE, pokud mají geometrie alespoň jeden společný bod, ale jejich vnitřní plochy se nepřekrývají

## ST\_Transform

- vrací novou geometrii transformovanou do zadaného souřadnicového systému

## ST\_Union

- vrací geometrii vzniklou sloučením jiných geometrií

## ST\_Within

- vrací hodnotu TRUE, pokud je geometrie kompletně uvnitř jiné geometrie

## ST\_X / ST\_Y

- vrací X / Y souřadnici zadaného bodu

# 5 Dotazy

## 5.1 Atributové

### 1) Kolik je v ČR fakultních nemocnic s výměrou větší než 10 ha?

```
SELECT COUNT(nazev_nemocnice) AS "pocet nemocnic"  
FROM nemocnice  
WHERE nazev_nemocnice LIKE '%akultn%' AND vymera_km2 > 0.1;
```

	pocet nemocnic
1	7

### 2) Jaké názvy obcí mají společné okresy České Budějovice a Strakonice?

```
SELECT nazev_obce AS obec  
FROM obce  
WHERE nazev_okresu = 'České Budějovice'  
INTERSECT  
SELECT nazev_obce  
FROM obce  
WHERE nazev_okresu = 'Strakonice';
```

	obec
1	Záboří
2	Radošovice
3	Nová Ves
4	Doubravice

### 3) Jaké obce Jihomoravského kraje s výměrou 14 - 15 km<sup>2</sup> mají kód okresu 3710, 3711 nebo 3712?

```
SELECT nazev_obce AS obec, kod_okresu AS "kod okresu", nazev_kraje AS kraj,
ROUND(CAST(vymera_km2 AS NUMERIC), 2) AS "vymera obce [km2]"
FROM obce
WHERE kod_okresu IN (3710, 3711, 3712) AND nazev_kraje = 'Jihomoravský' AND
vymera_km2 >= 14 AND vymera_km2 <= 15;
```

	obec	kod okresu	kraj	vymera obce [km2]
1	Slavkov u Brna	3712	Jihomoravský	14.95
2	Příštpo	3710	Jihomoravský	14.41
3	Břestek	3711	Jihomoravský	14.13
4	Orlovice	3712	Jihomoravský	14.45
5	Záhorovice	3711	Jihomoravský	14.91

### 4) Ve kterém kraji je největší okres? Uveďte kraj, okres a jeho rozlohu.

```
SELECT nazev_kraje AS kraj, nazev_okresu AS okres, ROUND(CAST(vymera_km2 AS
NUMERIC), 2) AS "vymera okresu [km2]"
FROM okresy
WHERE vymera_km2 = (SELECT MAX(vymera_km2)
FROM okresy);
```

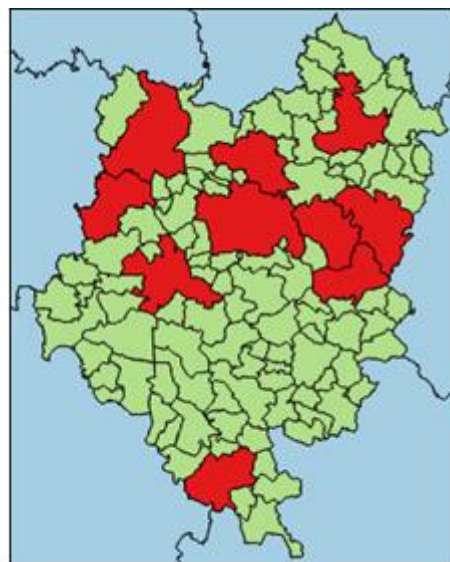
	kraj	okres	vymera okresu [km2]
1	Západočeský	Klatovy	1945.71

### 5) Kolik procent plochy okresu Tábor zaujímá jeho 10 největších obcí?

```
SELECT ROUND(CAST((100 * (SELECT SUM(vymera_km2)
FROM obce
WHERE nazev_obce IN (SELECT nazev_obce
FROM obce
WHERE nazev_okresu = 'Tábor'
ORDER BY vymera_km2 DESC
LIMIT 10)
AND nazev_okresu = 'Tábor'))
/ (SELECT vymera_km2
FROM okresy
WHERE nazev_okresu = 'Tábor'))
AS NUMERIC), 2) AS "podíl plochy [%]";
```



	podíl plochy [%]
1	28.60



**6) Ve kterých okresech je celková plocha obcí začínajících na písmeno P větší než 200 km<sup>2</sup> a kolik těch obcí je?**

```
SELECT nazev_okresu AS okres, ROUND(CAST(SUM(vymera_km2) AS NUMERIC), 2) AS
"vymera obcí na P [km2]", count(nazev_obce) AS "pocet obcí na P"
FROM (SELECT nazev_obce, nazev_okresu, vymera_km2
      FROM obce
      WHERE nazev_obce LIKE 'P%') AS obce_na_P
GROUP BY nazev_okresu
HAVING SUM(vymera_km2) >= 200
ORDER BY SUM(vymera_km2) DESC;
```

	okres	vymera obcí na P [km2]	pocet obcí na P
1	Hlavní město Praha	496.16	1
2	Klatovy	253.70	7
3	Louny	216.66	8
4	Písek	208.41	9
5	Český Krumlov	205.43	4

## 5.2 Prostorové

**7) Kolika obcemi prochází nějaká dálnice?**

```
SELECT count(obce.id) AS "pocet obcí"
FROM obce
JOIN silnice
ON ST_Intersects(silnice.geom, obce.geom)
WHERE druh_silnice = 'motorway';
```

	pocet obcí
1	2808

### 8) Kolik lékáren se nachází na území obce Prachatice?

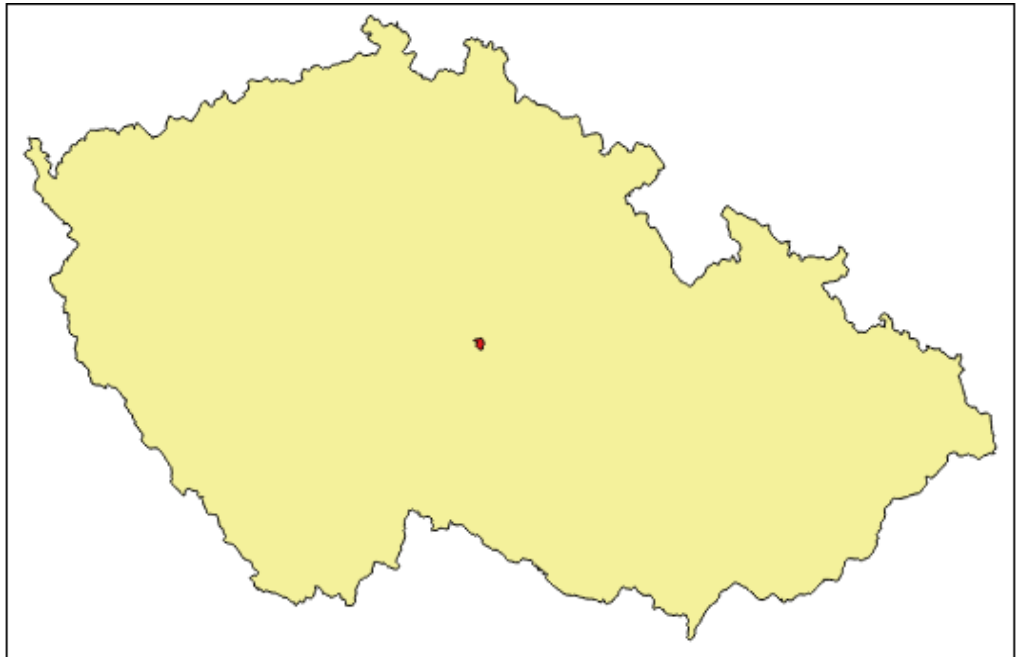
```
SELECT COUNT(*) AS "pocet lekaren"  
FROM lekarny  
JOIN (SELECT geom  
      FROM obce  
      WHERE nazev_obce= 'Prachatice') AS prachatice  
ON ST_Within(lekarny.geom, prachatice.geom);
```

	pocet lekaren
1	3

### 9) Na území které obce se nachází střed ČR?

```
SELECT nazev_obce AS obec  
FROM obce  
JOIN (SELECT ST_Centroid(cr.geom) AS geom  
      FROM (SELECT ST_Union(kraje.geom) AS geom  
            FROM kraje) AS cr)  
      AS stred  
ON ST_Within(stred.geom, obce.geom);
```

	obec
1	Čihošť



### 10) Který okres má nejdelší hranici?

```
SELECT nazev_okresu AS okres, ROUND(CAST(ST_Perimeter(okresy.geom) / 1000 AS  
NUMERIC), 2) AS "obvod [km]"  
FROM okresy  
GROUP BY nazev_okresu, okresy.geom  
ORDER BY "obvod [km]" DESC  
LIMIT 1;
```

	okres	obvod [km]
1	Brno-venkov	458.68

### 11) V kolika procentech obcí není žádná lékárna?

```
SELECT ROUND(CAST((100 * (SELECT COUNT(id)
                        FROM obce
                        WHERE id NOT IN (SELECT DISTINCT obce.id
                                      FROM obce
                                      JOIN lekarny
                                      ON ST_Within(lekarny.geom,
                                                  obce.geom)))
                        / (SELECT COUNT(id)
                          FROM obce))
              AS NUMERIC), 2) AS "podil obci bez lekaren [%]";
```

	podil obci bez lekaren [%]
1	94.00

### 12) Které okresy Jihomoravského kraje sousedí s krajem Severomoravským?

```
SELECT DISTINCT okresy_JM.nazev_okresu AS okres
FROM (SELECT * FROM okresy
      WHERE nazev_kraje LIKE 'Jihomor%') AS okresy_JM
JOIN (SELECT *
      FROM okresy
      WHERE nazev_kraje LIKE 'Severomor%') AS okresy_SM
ON ST_Touches(okresy_SM.geom, okresy_JM.geom);
```

	okres
1	Zlín
2	Prostějov
3	Kroměříž

### 13) Která lékárna je nejbližší středu Prahy?

```
SELECT nazev_lekarny AS lekarna, ROUND(CAST(ST_Distance(lekarny.geom,
ST_Centroid(praha.geom)) AS NUMERIC), 2) AS "vzdalenost [m]"
FROM lekarny
JOIN (SELECT geom
      FROM obce
      WHERE nazev_obce = 'Praha') AS praha
ON praha.geom && lekarny.geom
ORDER BY "vzdalenost [m]" ASC
LIMIT 1;
```

	lekarna	vzdalenost [m]
1	Dr. Max	260.52

#### 14) V jakých obcích jsou alespoň 4 lékárny Dr. MAX?

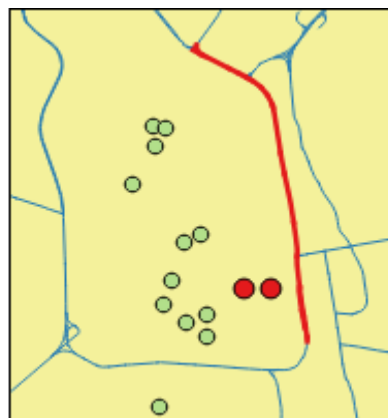
```
SELECT nazev_obce AS obec, COUNT(nazev_obce) AS "pocet lekaren Dr. MAX"
FROM obec
JOIN (SELECT geom
      FROM lekarny
      WHERE nazev_lekarny LIKE 'Dr.%' OR nazev_lekarny LIKE 'DR.%') AS dr
ON ST_Within(dr.geom, obec.geom)
GROUP BY nazev_obce
HAVING COUNT(nazev_obce) >= 4
ORDER BY COUNT(nazev_obce) DESC;
```

	obec	pocet lekaren Dr. MAX
1	Praha	33
2	Brno	13
3	Teplice	4
4	Ostrava	4
5	Frýdek-Místek	4

#### 15) Které lékárny jsou ve vzdálenosti do 0.5 km od ulice Nádražní v Českých Budějovicích?

```
SELECT DISTINCT nazev_lekarny AS lekarna
FROM lekarny
JOIN (SELECT nazev_silnice, silnice.geom
      FROM silnice
      JOIN obec
      ON ST_Within(silnice.geom, obec.geom)
      WHERE nazev_obce = 'České Budějovice') AS silnice_v_cb
ON ST_Within(lekarny.geom, ST_Buffer(silnice_v_cb.geom, 500))
WHERE nazev_silnice = 'Nádražní';
```

	lekarna
1	Lékárna Lannova třída
2	Lékárna Benu



### 16) Který helipad je nejbližší dálnici nebo silnici 1. třídy?

```
SELECT nazev_helipadu AS helipad, nazev_silnice AS silnice,
ROUND(CAST(ST_Distance(helipady.geom, silnice.geom) AS NUMERIC), 2) AS
"vzdalenost [m]"
FROM helipady
JOIN (SELECT geom, nazev_silnice
      FROM silnice
      WHERE druh_silnice = 'motorway' OR druh_silnice = 'primary') AS silnice
ON helipady.geom && silnice.geom
ORDER BY "vzdalenost [m]"
LIMIT 1;
```

	helipad	silnice	vzdalenost [m]
1	Nemocnice Frýdlant - přistávací plocha	Fügnerova	30.09

### 17) Kolik metrů má obvod nejvýhodněji položené nemocnice ve Středočeském kraji?

```
SELECT nazev_nemocnice AS nemocnice, ROUND(CAST(ST_Perimeter(nemocnice.geom)
AS NUMERIC), 2) AS "obvod [m]" FROM nemocnice
WHERE id IN (SELECT id
             FROM (SELECT nemocnice.id, nazev_nemocnice,
ST_Centroid(nemocnice.geom) AS C, nemocnice.geom
                 FROM nemocnice
                 JOIN kraje
                 ON ST_Intersects(nemocnice.geom, kraje.geom)
                 WHERE nazev_kraje LIKE 'Středočes%') AS nem_v_JC
             ORDER BY ST_X(C) DESC
             LIMIT 1);
```

	nemocnice	obvod [m]
1	Městská nemocnice Čáslav	633.64

### 18) Ve kterých okresech je více lékáren než obcí?

```
SELECT a.nazev_okresu AS okres, l AS "pocet lekaren", o AS "pocet obci"
FROM (SELECT nazev_okresu, COUNT(lekarny.id) AS l
      FROM lekarny
      JOIN okresy
      ON ST_Within(lekarny.geom, okresy.geom)
      GROUP BY nazev_okresu) AS a
JOIN (SELECT okresy.nazev_okresu, COUNT(obce.id) AS o
      FROM obce
      JOIN okresy
      ON ST_Intersects(obce.geom, okresy.geom) GROUP BY okresy.nazev_okresu)
AS b
ON a.nazev_okresu = b.nazev_okresu
WHERE l > o;
```

	okres	pocet lekaren	pocet obci
1	Brno-město	106	29
2	Hlavní město Praha	249	40

## 19) Jak velké procento helipadů v Severočeském kraji je v prostorách nemocnice?

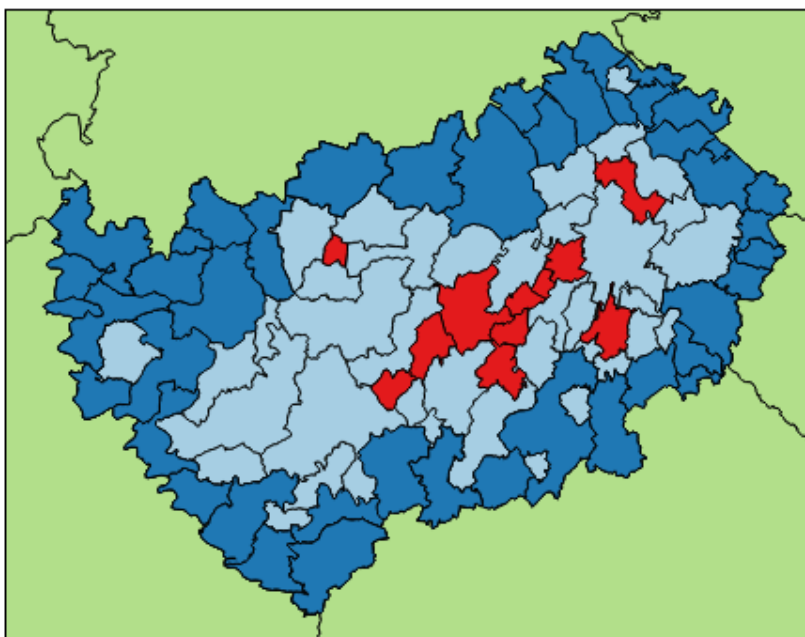
```
SELECT ROUND(CAST((100 * (SELECT count(DISTINCT helipady.id)
FROM helipady
JOIN (SELECT nemocnice.geom
FROM nemocnice
JOIN kraje
ON ST_Within(nemocnice.geom, kraje.geom)
WHERE nazev_kraje LIKE 'Severoč%') AS nem_v_SC
ON ST_Dwithin(helipady.geom, nem_v_SC.geom, 100))
/ (SELECT count(helipady.id)
FROM helipady
JOIN kraje
ON ST_Within(helipady.geom, kraje.geom)
WHERE nazev_kraje LIKE 'Severoč%'))
AS NUMERIC), 2) AS "helipady v nemocnicich [%]";
```

	helipady v nemocnicich [%]
1	53.00

## 20) Které obce okresu Kutná hora nesousedí s obcemi, které jsou na hranici tohoto okresu (uvnitř okresu)?

```
SELECT o.nazev_obce AS obec
FROM (SELECT nazev_obce, geom
FROM obce
WHERE nazev_okresu = 'Kutná Hora') AS o
JOIN (SELECT ST_Union(obce_hr.geom) AS geom
FROM (SELECT obce_KH.geom
FROM (SELECT geom
FROM obce
WHERE nazev_okresu = 'Kutná Hora') AS obce_KH
JOIN (SELECT geom
FROM obce
WHERE nazev_okresu != 'Kutná Hora') AS obce_ne_KH
ON ST_Touches(obce_KH.geom, obce_ne_KH.geom)) AS obce_hr)
AS obce_hranice
ON ST_Disjoint(o.geom, obce_hranice.geom);
```

	obec
1	Košice
2	Močovice
3	Souňov
4	Úmonín
5	Chotusice
6	Třebonín
7	Paběnice
8	Opatovice I
9	Vodranty
10	Štipoklas
11	Tupadly



## 6 Závěr

Dohromady bylo vytvořeno 7 datových vrstev, na které odkazuje celkově 20 SQL dotazů. Dále byla vytvořena dokumentace popisující nově vytvořené datové vrstvy a jejich tvorbu, SQL dotazy i několik informací k jednotlivým zdrojům.

Bohužel poskytnutá data často nejsou aktuální (například ve městě Prachatice, které má cca 12 000 obyvatel jsou údajně pouze 3 lékárny, jelikož odsud pocházím, vím, že lékáren je zde nejméně 6). Dále u některých vrstev (např. silnice) chybí označení všech dálnic a některých silnic I. a II. třídy.

## 7 Přílohy

Odevzdané soubory – uzpd\_2015\_c.zip

- dokumentace.pdf
- prezentace.pdf
- davka.sql

## 8 Zdroje

[http://wiki.openstreetmap.org/wiki/Cs:Map\\_Features](http://wiki.openstreetmap.org/wiki/Cs:Map_Features)

<https://cs.wikipedia.org/wiki/OpenStreetMap>

<http://www.openstreetmap.org/about>

<http://www.geobusiness.cz/2011/04/co-je-to-ruian/>

[http://postgis.net/docs/reference.html#Spatial\\_Relationships\\_Measurements](http://postgis.net/docs/reference.html#Spatial_Relationships_Measurements)