

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

FAKULTA STAVEBNÍ

OBOR GEOMATIKA

## SEMESTRÁLNÍ PRÁCE

TVORBA TÉMATICKÝCH VRSTEV, VALIDACE DAT A VYTVOŘENÍ  
TUTORIÁLU



Vedoucí práce: Ing. Martin Landa Ph.D.

Katedra Geomatiky (K155)

únor 2017

DAVID TĚTHAL, PETR DYRC

## ZADÁNÍ

Navrhněte a vytvořte tématické vrstvy (např. vodní toky, vodní plochy, lesy, silnice, železnice a pod.) na základě dat OpenStreetMap a další otevřených zdrojů.

Aplikujte testy datové integrity a odstraňte případné nekonzistence v datech.

Vytvořte tutoriál - tj. sadu atributových a prostorových dotazů nad databází.

1. ÚVOD.....	4
1.1 VÝBĚR TÉMATU.....	4
1.2 POUŽITÝ SOFTWARE.....	4
2. ZDROJE DAT.....	5
2.1 ČSÚ .....	5
2.2 RÚIAN.....	5
3. TÉMATICKÉ VRSTVY .....	6
3.1 cesko .....	6
3.2 kraje.....	7
3.3 obce .....	8
3.4 sldb.....	8
3.5 strany.....	9
3.6 kraje2strany .....	11
4. APLIKOVANÉ FUNKCE.....	13
5. DOTAZY .....	14
5.1 ATRIBUTOVÉ .....	14
ZÁVĚR.....	19
REFERENCE.....	20

# 1. ÚVOD

## 1.1 VÝBĚR TÉMATU

Pro náš semestrální projekt jsme se rozhodli pracovat s otevřenými daty, které poskytuje Český statistický úřad (dále ČSÚ) a jejich následným propojením s prostorovými daty z Registru územní identifikace, adres a nemovitostí (dále RÚIAN). Z dat ČSÚ jsme si vybrali především informace o sčítání lidu a z krajských voleb, které jsme pro svou potřebu zpracovali tak, aby je bylo možno vhodně uložit do pracovní databáze. Nad těmito daty jsme následně vytvořili, jak atributové, tak prostorové dotazy.

## 1.2 POUŽITÝ SOFTWARE

### PostGIS

Jedná se o nadstavbu pro objektově-relační databázový systém PostgreSQL, která přidává podporu pro geografické objekty [1]. Jedná se o volně dostupný software.

### QGIS

Konkrétně se jedná o verzi 2.18.3 Las Palmas. QGIS je volně dostupný a multiplatformní geografický informační systém (GIS) [2].

### PyCharm

Konkrétně se jedná o verzi 4.0.5. PyCharm je multiplatformní vývojové prostředí používané pro počítačové programování specializující se na jazyk Python. PyCharm byl vyvinut českou firmou JetBrains [3].

## 2. ZDROJE DAT

### 2.1 ČSÚ

ČSÚ je ústřední orgán státní správy České republiky. Byl zřízen dne 8. ledna 1969. Jako hlavní orgán státní statistické služby koordinuje sběr a zpracování statistických údajů, které provádějí jednotlivá ministerstva [4]. Český statistický úřad na základě vládního Akčního plánu "Partnerství pro otevřené vládnutí", zveřejňuje dostupná data z výsledků voleb v otevřených formátech zároveň s textovým popisem obsahu zveřejněných dat [5].

Při sběru dat jsme se zaměřovali pouze na otevřená data, poskytovaná přímo na webových stránkách ČSÚ. Konkrétně data ze sčítání lidu a z krajských voleb. Co se týče stáří dat, tak zde nastal problém s konzistencí dat, kdy pro volební výsledky jsme se snažili najít co možná nejaktuálnější data tj. Volby zastupitelstev krajů 2016. Data pro sčítání lidu jsou však z roku 2011. Vzhledem ke skutečnosti, že se sčítání lidu provádí pouze jednou za deset let se s touto realitou nedá nic dělat.

ČSÚ svá data poskytuje ve více formátech. V nabídce jsou formáty DBF, XML, TSV, XLS a CSV. Pro data ze sčítání lidu jsme použili formát CSV (Comma-separated values) což je jednoduchý souborový formát určený pro výměnu tabulkových dat [6]. Data z krajských voleb je možné získat pouze ve formátu XML (Extensible Markup Language), kdy celý dokument je definován pomocí kaskádových stylů.

### 2.2 RÚIAN

RÚIAN je jedním ze čtyř základních registrů dle zákona č. 111/2009 Sb. Tyto registry jsou základním stavebním kamenem elektronizace veřejné správy České republiky. Hlavní funkcí registru RÚIAN je poskytování informačních a servisních služeb pro veřejnost a veřejnou správu. Registr územní identifikace je postaven na již vytvořeném Informačním systému katastru nemovitostí, provozovaném Českým úřadem zeměměřickým a katastrálním [7].

Vzhledem k charakteru dat z ČSÚ jsme si z RÚIAN vybrali pouze takové vrstvy, které vhodně slouží k reprezentaci informací z tzv. tabulkových dat poskytovaných ČSÚ. Dá se říct, že data RÚIAN jsou použita převážně pro prostorovou informaci na rozdíl od dat z ČSÚ, která nabízejí informaci pouze popisnou a sama o sobě by se dala použít jen pro atributové dotazy.

Ze schématu RÚIAN byla použita polygonová vrstva *staty*, což je vrstva obsahující pouze jeden polygon celé České republiky. Dále byla použita vrstva *vusc*, která obsahuje polygony krajů a jako poslední vrstva byla zvolena vrstva *obce*, obsahující polygony obcí a další popisné informace.

### 3. TÉMATICKÉ VRSTVY

#### 3.1 cesko

Jednou vrstvou ze schématu RÚIAN je vrstva *cesko*, která obsahuje jeden prvek a to polygon České republiky. Geometrie je uložena ve sloupci *geom* jako multipolygon v souřadnicovém systému Krovak East North (5514). Dále tabulka obsahuje sloupce *nazev* a *kod*.

Vrstva byla do schématu c16 importována ze schématu ruian:

```
CREATE TABLE cesko AS
SELECT nazev, kod, geom
FROM staty
```

Dále byl přidán primární klíč nad sloupcem *kod*:

```
ALTER TABLE cesko ADD PRIMARY KEY (kod);
```

A nad sloupcem *geom* byl vytvořen prostorový index:

```
CREATE index sidx_cesko_index ON cesko(geom);
```

Pro zachování konzistence dat a odstranění chyb u polygonových prvků bylo nutné provést validitu jejich geometrie:

```
SELECT kod FROM cesko WHERE not st_isvalid(geom);
```

Při zjišťování validity nebyly nalezeny žádné chyby.

### 3.2 kraje

Vrstva *kraje* obsahuje geometrii jednotlivých polygonů pro kraje v České republice. Mimo geometrie uložené ve sloupci *geom* jako multipolygon v souřadnicovém systému Krovak East North (5514) tabulka obsahovala ještě sloupce *nazev* a *kod*.

Vrstva *kraje* byla do schématu *c16* importována ze schématu *ruian* z vrstvy *vusc*:

```
CREATE TABLE kraje AS
SELECT kod, geom
FROM vusc
```

Jelikož kódy u jednotlivých krajů neodpovídaly kódování, používaném v ČSÚ, bylo nutné je změnit:

```
UPDATE kraje SET kod = 3018 WHERE kod = 19;
UPDATE kraje SET kod = 3026 WHERE kod = 27;
UPDATE kraje SET kod = 3034 WHERE kod = 35;
UPDATE kraje SET kod = 3042 WHERE kod = 43;
UPDATE kraje SET kod = 3051 WHERE kod = 51;
UPDATE kraje SET kod = 3069 WHERE kod = 60;
UPDATE kraje SET kod = 3077 WHERE kod = 78;
UPDATE kraje SET kod = 3085 WHERE kod = 86;
UPDATE kraje SET kod = 3093 WHERE kod = 94;
UPDATE kraje SET kod = 3107 WHERE kod = 108;
UPDATE kraje SET kod = 3115 WHERE kod = 116;
UPDATE kraje SET kod = 3123 WHERE kod = 124;
UPDATE kraje SET kod = 3131 WHERE kod = 141;
UPDATE kraje SET kod = 3140 WHERE kod = 132;
```

Dále byl přidán primární klíč nad sloupcem *kod*:

```
ALTER TABLE kraje ADD PRIMARY KEY (kod);
```

A nad sloupcem *geom* byl vytvořen prostorový index:

```
CREATE index kraje_index ON kraje(geom)
```

Stejně jako v předchozí vrstvě i zde byla ověřena validita sloupce obsahujícího geometrii prvku:

```
SELECT kod FROM kraje WHERE not st_isvalid(geom);
```

Při zjišťování validity vrstvy *kraje* nebyly nalezeny žádné chyby.

### 3.3 obce

Vrstva *obce* obsahuje pouze geometrii obcí v České republice. Stejně jako v předchozím schématu je geometrie uložena ve sloupci *geom* jako multipolygon v souřadnicovém systému Krovak East North (5514). V tabulce *obce* je ještě jejich kód, který odpovídá kódování ČSÚ.

Vrstva *obce* byla do schématu *c16* importována ze schématu *ruian* z vrstvy *obce*:

```
CREATE TABLE obce AS
SELECT kod, geom
FROM obce
```

Primární klíč byl vytvořen nad sloupcem *kod*:

```
ALTER TABLE obce ADD PRIMARY KEY (kod);
```

A nad sloupcem *geom* byl vytvořen prostorový index:

```
CREATE index obce_index ON obce(geom)
```

Na závěr byla ověřena validita sloupce obsahujícího geometrii prvku:

```
SELECT kod FROM obce WHERE not st_isvalid(geom);
```

Při zjišťování validity vrstvy *obce* nebyly nalezeny žádné chyby.

Vzhledem k realitě, že všechny tabulky obsahující geometrii jsou z jednoho zdroje a tedy mají stejný souřadnicový systém, nebyla jejich transformace nutná.

### 3.4 sldb

Tabulka *sldb* byla vytvořena z dat českého statistického úřadu a proto neobsahuje žádnou geometrii. V atributové tabulce *sldb* se nacházejí informace ze sčítání lidu v roce 2011 pro všechny obce, okresy, kraje a celou Českou republiku.

Data ze sčítání lidu byla získána ve formátu CSV (popsaném v kapitole 2.1). Pro import do databáze bylo nutné nejprve vytvořit dávkový soubor s příponou „.sql“ obsahující soubor SQL příkazů, které vytvoří příslušnou tabulku a data do ní zapíše.

Dávkový soubor byl vytvořen pomocí funkce *ogr2ogr* následovně:

```
ogr2ogr -f PGDump sldb.sql -oo AUTODETECT_TYPE=YES -lco SCHEMA=c16 sldb.csv sldb
```



Při vytváření souboru `sldb.sql` bylo nutné automaticky detekovat kódování řetězců znaků původního souboru. Bylo nutné zachovat původní kódování `windows-1250`. Pokud by tak nebylo učiněno, kódování by se změnilo na `UTF-8` a názvy uložené v tabulce by byly špatně.

Soubor `sldb.sql` byl posléze lehce upraven, kdy názvy sloupců při vytváření tabulky byly přejmenovány tak, aby bylo na první pohled jasné, co obsahují. Výsledný soubor byl nahrán do schématu `c16` příkazem:

```
psql pgis_uzpd -h geo102.fsv.cvut.cz -U uzpd_c16 -f sldb.sql
```

V posledním kroku byl vytvořen primární klíč nad sloupcem `ogc_fid`:

```
ALTER TABLE sldb ADD PRIMARY KEY (ogc_fid);
```

### 3.5 strany

Tabulka *strany* obsahuje všechny politické strany resp. jejich název a identifikační číslo přidělené ČSÚ, zvolené alespoň v jednom kraji při krajských volbách v roce 2016.

Informace z Krajských voleb 2016 poskytuje ČSÚ ve formátu XML [8]. Pro zjištění politických stran byly všechny strany z volebních výsledků pro jednotlivé kraje uloženy do přechodné tabulky. Z té byla posléze vytvořena výsledná tabulka, ve které již nevznikaly duplicity jednotlivých stran (jedna strana mohla být volena ve více krajích).

Pro čtení formátu XML byl napsán skript v programu PyCharm, který ze souboru nabízeného ČSÚ vybral pouze námi požadovaná data, ze kterých byla vytvořena SQL dávka vypsaná do konzole a posléze uložena do souboru s příponou „.sql“. Dávkový soubor obsahoval příkazy pro vytvoření přechodné a finální tabulky spolu s vytvořením primárního klíče a odstraněním přechodné tabulky. Po nahrání dávkového souboru byla tedy vytvořena tabulka *strany*, která obsahovala primární klíč a byla bez duplicit.

Skript v jazyce python generující dávkový soubor:

```
from lxml import etree
import os

fname = raw_input("File name:")

if os.path.exists(fname) != True:
    print("Open Failed")
else:
```

```

print "\nInput file: ", fname

xmldoc = etree.parse(fname).getroot()
print "CREATE TABLE strany_all (id INT NOT NULL, jmeno CHAR(60)); \n"

for kraj in xmldoc.findall('KRZAST'):

    for strana in kraj.findall('STRANA'):
        cislo = strana.get('KSTRANA')
        jmeno = strana.get('NAZ_STR')

        print "INSERT INTO strany_all (id, jmeno)"
        print "VALUES (", cislo, ", ", jmeno, ");"

print "CREATE TABLE strany AS SELECT DISTINCT id, jmeno FROM strany_all ORDER BY id ASC; \n"
print "DROP TABLE strany_all; \n"
print "ALTER TABLE strany ADD PRIMARY KEY (id);"

task = raw_input("Print to file (y/n): ")
if task == "y":
    file_name = raw_input("Enter name of text file: ") + '.sql'
    if os.path.exists(file_name) == True:
        print "File already exists"
        file_ex = raw_input("Replace existing file? (y/n): ")
    else:
        file_ex = "y"
    if file_ex == "y":
        with open(file_name, 'w') as new_file:

            new_file.write("CREATE TABLE strany_all (id INT NOT NULL UNIQUE, jmeno CHAR(60)); \n")

            for kraj in xmldoc.findall('KRZAST'):

                for strana in kraj.findall('STRANA'):
                    cislo = strana.get('KSTRANA')
                    jmeno = strana.get('NAZ_STR')

                    new_file.write("INSERT INTO strany_all (id, jmeno) \n")
                    new_file.write("VALUES (%s, '%s'); \n" % (cislo, jmeno.encode('utf-8')))

            new_file.write("CREATE TABLE strany AS SELECT DISTINCT id, jmeno FROM strany_all ORDER
BY id ASC; \n")
            new_file.write("DROP TABLE strany_all; \n")
            new_file.write("ALTER TABLE strany ADD PRIMARY KEY (id);")

    new_file.close()

```

Dávkový soubor byl načten stejně jako v případě tabulky sldb příkazem:

```
psql pgis_uzpd -h geo102.fsv.cvut.cz -U uzpd_c16 -f strany_davka.sql
```

### 3.6 kraje2strany

Tabulka *kraje2strany* je vazební tabulka, která ke každému kraji udává strany, které v něm byly voleny (sloupce *kraje\_id* a *strana\_id*). Mimo této vazby také přidává každé volené straně pro určitý kraj statistiky, a to celkový počet hlasů a procento hlasů (sloupce *hlasy* a *proc\_hlasu*). Jako poslední sloupec obsahuje identifikační číslo jednotlivých záznamů (sloupec *id*), to slouží jako primární klíč.

Stejně jako v případě v kapitole 3.5, tak i zde bylo čerpáno ze stejného XML souboru. V tomto případě bylo možno tabulku vytvořit přímo bez použití dočasné tabulky. Jednotlivé záznamy byly získány z XML pomocí skriptu v programu PyCharm, identifikační číslo bylo generováno automaticky. Dávkový soubor obsahuje rovněž vytvoření primárního klíče, proto další operace po jeho nahrání již nejsou nutné.

Skript se od předchozího až na rozdílné informace, které jsou čerpány z XML souboru, moc neliší:

```
from lxml import etree
import os

fname = raw_input("File name:")

if os.path.exists(fname) != True:
    print("Open Failed")
else:
    print "\nInput file: ", fname

xmldoc = etree.parse(fname).getroot()
id= 0;

print "CREATE TABLE kraje2strany (kraje_id int not null unique, strana_id int, hlasy int, proc_hlasu float);"
```

```

for okres in xmlDoc.findall('KRZAST'):
    cislo_okresu = okres.get('CIS_KRZAST')

    for strana in okres.findall('STRANA'):
        id_strana = strana.get('KSTRANA')

        for data in strana.findall('HODNOTY_STRANA'):
            hlasy = data.get("HLASY")
            procento = data.get("PROC_HLASU")

            print "INSERT INTO kraje2strany (kraje_id, strana_id, hlasy, proc_hlasu)"
            print "VALUES (" ,id_strana," ,",cislo_okresu," ,",hlasy," ,",procento,");"

task = raw_input("Print to file (y/n): ")
if task == "y":
    file_name = raw_input("Enter name of text file: ") + '.sql'
    if os.path.exists(file_name) == True:
        print "File already exists"
        file_ex = raw_input("Replace existing file? (y/n): ")
    else:
        file_ex = "y"
    if file_ex == "y":
        with open(file_name, 'w') as new_file:

            new_file.write("CREATE TABLE kraje2strany (id INT NOT NULL UNIQUE PRIMARY KEY, kraje_id
INT, strana_id INT, hlasy INT, proc_hlasu FLOAT); \n")

            for okres in xmlDoc.findall('KRZAST'):
                cislo_okresu = okres.get('CIS_KRZAST')

                for strana in okres.findall('STRANA'):
                    id_strana = strana.get('KSTRANA')

                    for data in strana.findall('HODNOTY_STRANA'):
                        hlasy = data.get("HLASY")
                        procento = data.get("PROC_HLASU")
                        id += 1

                        new_file.write("INSERT INTO kraje2strany (id, kraje_id, strana_id, hlasy, proc_hlasu) \n")
                        new_file.write("VALUES (%s, %s , %s , %s , %s ); \n" % (id, cislo_okresu, id_strana, hlasy,
procento))

            new_file.close()

```

## 4. APLIKOVANÉ FUNKCE

Následující výčet stručný popis funkcí použitých při tvorbě prostorových dotazů [9].

**ST\_Area**

- vrátí výměru, pokud je geometrie polygon nebo multipolygon\*

**ST\_Perimeter**

- vrátí délku hranice prvku (polygon, multipolygon)\*

**ST\_Within**

- vrací hodnotu TRUE pokud je geometrie A kompletně uvnitř geometrie B

**ST\_Boundary**

- vrací okraj geometrie

**ST\_Intersects**

- vrací hodnotu TRUE pokud se geometrie A překrývá alespoň částí geometrii B

**ST\_Distance**

- vrací 2D Kartézskou vzdálenost mezi dvěma geometriemi\*

\*jednotky na výstupu jsou totožné s původními jednotkami geometrie

## 5. DOTAZY

Kapitola DOTAZY obsahuje SQL Tutoriál složený s atributových a prostorových dotazů nad předem vytvořenými schématy.

### 5.1 ATRIBUTOVÉ

Jaký je kraj s nejmenší volební účastí v Krajských volbách za rok 2016:

Ústecký kraj (23%)

```
select distinct nazev, sum(hlasy)/obyv_celk volebni_ucast
from kraje2strany k
join sldb s on k.kraje_id = s.uzkod
where uzcis = 100
group by nazev, obyv_celk
order by volebni_ucast asc limit 1
```

Jaké jsou tři nejvolenější strany v Krajských volbách pro rok 2016:

ANO 2011, ČSSD, KSČM

```
select jmeno, sum(hlasy) pocet_hlasu
from strany s
join kraje2strany k on s.id = k.strana_id
group by jmeno
order by pocet_hlasu desc limit 3
```

Jaká je obec s největším a nejmenším počtem obyvatel na jeden byt:

Březina (1.5 obyvatel)

Všehrady (14.83 obyvatel)

```
(select nazev, obyv_celk/obydlene_byty obyv_byt
from sldb
where uzcis = 43
order by obyv_byt asc limit 1)
union
(select nazev, obyv_celk/obydlene_byty obyv_byt
from sldb
where uzcis = 43
order by obyv_byt desc limit 1)
```

Jaké je procento mužů a žen v ČR:

49% mužů, 51% žen

```
select round(cast(muzi/obyv_celk as numeric), 2) procento_muzu,  
round(cast(zeny/obyv_celk as numeric), 2) procento_zen  
from sldb  
where uzcis = 97
```

V jakých krajích získala Pirátská strana více než 5% hlasů a kolik to bylo:

Karlovarský kraj (5.46%)

```
select nazev, proc_hlasu  
from sldb s  
join kraje2strany k on s.uzkod = k.kraje_id  
join strany st on st.id = k.strana_id  
where uzcis = 100 and proc_hlasu > 5 and jmeno = 'Česká pirátská strana'
```

Jak se jmenuje obec s nejmenším počtem obyvatel v ČR:

Březina (3 obyvatel)

```
select nazev, obyv_celk pocet_obyvatel  
from sldb  
order by obyv_celk asc limit 1
```

Jak se jmenuje obec s největší mírou nezaměstnanosti v ČR:

Bílá Voda (56%)

```
select nazev, ((ekonomicky_aktivni - ekonomicky_aktivni_zamestnani)/(ekonomicky_aktivni))  
mira_nez  
from sldb  
where uzcis = 43  
order by mira_nez desc limit 1
```

Jaká strana je volena v kraji, kde je největší procento obyvatel nad 65 let:

ANO 2011

```
select st.jmeno
from strany st
join kraje2strany ks on st.id = ks.strana_id
join sldb s on ks.kraje_id = s.uzkod
where uzkod = (
    select uzkod
    from sldb
    where uzcis = 100
    order by over65/obyv_celk desc limit 1)
order by hlasy desc limit 1
```

Jak se jmenují obce s nejmenší mírou nezaměstnanosti v ČR:

Líšná, Račice, Kyjov, Březina, Kubova Huť, Vrbice, Třeštice, Jedlany, Kunčina Ves, Zvolenovice, Báčovice, Únice, Dobrá Voda u Pacova, Dubovice, Oplany, Vísky, Eš, Kratušín, Hracholusky, Lesná, Sekeřice, Pustá Rybná, Bratronice, Záhoří, Sedliště, Kostelec, Bečice, Bořetice, Pšánky, Horní Paseka, Minice, Brdy, Vysoká Lhota, Bludov, Zábrdí, Hornosín, Slověnice, Nicov, Rosička, Mlýnské Struhadlo, Kaničky, Řemíčov, Chleby, Krasíkovice, Borovnice, Pivkovice, Říkonín

```
select nazev, ((ekonomicky_aktivni - ekonomicky_aktivni_zamestnani)/(ekonomicky_aktivni))
mira_nez
from sldb
where uzcis = 43
order by mira_nez asc limit (
    select count(*) from sldb where ((ekonomicky_aktivni -
ekonomicky_aktivni_zamestnani)/(ekonomicky_aktivni)) =
(select ((ekonomicky_aktivni -
ekonomicky_aktivni_zamestnani)/(ekonomicky_aktivni)) mira_nez
from sldb
where uzcis = 43
order by mira_nez asc limit 1))
```



## 5.2 PROSTOROVÉ

Jaká je hustota obyvatel v Praze na km<sup>2</sup>:

2554.48 obyvatel/ km<sup>2</sup>

```
select obyv_celk/(st_area(geom)/1000000) hustota_obyvatel
from kraje k
join sldb s on k.kod = s.uzkod
where s.nazev = 'Hlavní město Praha'
```

Kolik obcí je v kraji s největším obvodem:

973

```
select count(*) pocet_obci
from obce o
join (select geom
      from kraje
      order by st_perimeter(geom) desc limit 1) k
on st_within(o.geom, k.geom)
```

Jaká je průměrná míra nezaměstnanosti obcí na hranici ČR:

14.9%

```
WITH cesko as
(SELECT st_boundary(geom) geom FROM cesko)
SELECT avg((ekonomicky_aktivni - ekonomicky_aktivni_zamestnani)/(ekonomicky_aktivni))
mira_nezam
FROM obce o
JOIN cesko c ON st_intersects(o.geom, c.geom)
JOIN sldb s ON o.kod = s.uzkod
```

Jaká je vzdálenost mezi obcemi s největším poměrem žen a mužů:

237.94 km

```
select st_distance(geom,(select geom
from obce o
join sldb s on o.kod = s.uzkod
order by obyv_celk/muzi desc limit 1))/1000 vzdalenost
from obce o
join sldb s on o.kod = s.uzkod
order by obyv_celk/zeny desc limit 1
```

Jaká je nejmenší obec v ČR:

Závist (0.42 km<sup>2</sup>)

```
select s.nazev, st_area(geom)/1000000 rozloha
from sldb s
join obce o on s.uzkod = o.kod
order by st_area(geom) asc limit 1
```

Jak se jmenuje obec s největší hustotou obyvatel nad 65 let (nejstarší obec):

Strukov

```
select s.nazev
from obce o
join sldb s on o.kod = s.uzkod
where uzcis = 43 and st_area(geom) is not NULL
order by (over65/obyv_celk)/st_area(geom) desc limit 1
```

Jaká je obec s nejmenší hustotou zalidnění v ČR:

Březina (1.89 obyvk/km<sup>2</sup>)

```
select s.nazev, obyv_celk/st_area(geom)/1000000 hustota_zalidneni
from sldb s
join obce o on s.uzkod = o.kod
order by obyv_celk/(st_area(geom)/1000000) asc limit 1
```

## ZÁVĚR

Cílem této práce bylo vytvořit tématické vrstvy z volně dostupných dat, jejichž prostorové prvky budou validní a dále nad vybraným schématem provést atributové a prostorové dotazy. K naší práci byla vybrána data z ČSÚ a RÚIAN. Práce s daty ČSÚ je díky přiložené dokumentaci snadná a přehledná a vzhledem k velkému množství nabízených formátů není problém s daty manipulovat. V práci je mimo jiné použito více možností k nahrání dat do databáze, které se liší původem a formátem dat. V semestrálním projektu se nám podařilo propojit atributová data z ČSÚ s prostorovými daty z RÚIAN.

Výsledný tutorial, je možné použít pro zjištění základních informací při psaní článků o volbách, při tvorbě přehledů, nebo pro další demografické analýzy na základě dat ze sčítání lidu.

Semestrální projekt by mohl být obohacen dalšími statistickými daty, které by zvýšily potenciál celého tutoriálu. Vhodným způsobem jak práci vylepšit by také mohla být vizualizace statistických dat společně s prostorovými ve formě kartodiagramů, nebo tématických map. Tématické mapy s volebními výsledky jsou velice populární ve všech médiích.

## REFERENCE

- [1] Wikipedia. [Online]. Available: <https://cs.wikipedia.org/wiki/PostGIS>
- [2] Wikipedia. [Online]. Available: <https://cs.wikipedia.org/wiki/QGIS>
- [3] Wikipedia. [Online]. Available: <https://en.wikipedia.org/wiki/PyCharm>
- [4] Wikipedia. [Online]. Available: [https://cs.wikipedia.org/wiki/wiki/Český\\_statistický\\_úřad](https://cs.wikipedia.org/wiki/wiki/Český_statistický_úřad)
- [5] Wikipedia. [Online]. Available: [https://www.czso.cz/csu/czso/otevrena\\_data](https://www.czso.cz/csu/czso/otevrena_data)
- [6] Wikipedia. [Online]. Available: <https://cs.wikipedia.org/wiki/CSV>
- [7] Geo-business. [Online]. Available: <http://www.geobusiness.cz/2011/04/co-je-to-ruian/>
- [8] Český statistický úřad, [Online]. Available: <http://volby.cz/pls/kz2016/vysledky>
- [9] PostGIS. [Online]. Available: <http://postgis.net/docs/>