

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

Fakulta stavební

Katedra geomatiky



DOKUMENTACE

ÚVOD DO ZPRACOVÁNÍ PROSTOROVÝCH DAT

Semestrální projekt

Skupina c18

Petra Jílková

ZS 2018/2019

OBSAH

1 ÚVOD	1
2 TVORBA VRSTEV	2
2.1 Přehled vrstev	2
2.2 Postup vytvoření	2
2.2.1 Vlastní vrstvy	2
2.2.2 Obce 3. fáze	3
2.2.3 Komunikace a budovy	4
2.2.4 Zastavěné části	4
2.3 Validace vrstev	6
3 ANALÝZA	7
3.1 Výpočet ukazatelů	7
3.1.1 Počet zastavěných částí v obci	7
3.1.2 Průměrná vzdálenost zastavěných částí	8
3.1.3 Délka komunikací v zastavěné části obce	8
3.2 Kategorizace obcí	9
4 PROSTOROVÉ A ATRIBUTOVÉ DOTAZY	11
5 ZÁVĚR	15
Použitý software	15
Reference	15

1 ÚVOD

Cílem semestrálního projektu předmětu Úvod do zpracování prostorových dat je vypracování analýzy vybraných obcí ČR pro účely terénního šetření, které probíhá v rámci projektu „Mapování infrastruktury a služeb venkova ČR“. Tento projekt je realizován Sdružením místních samospráv ČR (SMS ČR) za podpory Ministerstva pro místní rozvoj ČR a Ministerstva školství, mládeže a tělovýchovy. Hlavním cílem projektu „Mapování infrastruktury a služeb venkova ČR“ je shromáždění aktuálních informací o stavu infrastruktury a služeb obcí do 10 000 obyvatel, které budou sloužit jako podklad pro státní aparát k efektivnějšímu zaměření podpory z veřejných rozpočtů. V rámci terénního šetření probíhá v každé obci hodnocení stavu místních komunikací a mapování vybraných zájmových bodů (např. obecní úřad, pošta, sportoviště, obchod, sociální služby apod.), u kterých jsou hodnoceny atributy o stavu objektu či možnosti bezbariérového přístupu a pořizovány fotografie, které dokládají aktuální stav objektu. Hlavním výstupem projektu je pak Analytický geografický systém SMS ČR (AGIS SMS ČR), který kombinuje data z mapování s daty ze státních registrů a databází, jako např. RUIAN, NPÚ či AOPK. Tento systém by měl sloužit nejen starostům obcí, ale především orgánům státní správy na všech úrovních samosprávy.

Od března letošního roku bude probíhat již 3. fáze terénního šetření obcí, kdy budou mapovány obce mezi 3 000 a 10 000 obyvateli. V rámci příprav je nutné alespoň rámcově stanovit náročnost mapovaných obcí a vymežit rozsah mapovacích oblastí pro jednotlivé terénní pracovníky. S ohledem na průběh terénního mapování je náročnost zpracování obce posuzována dle následujících kritérií:

- počet obyvatel
- počet zastavěných částí
- délka komunikací v zastavěných částech
- průměrná vzdálenost zastavěných částí v katastru obce

V rámci semestrálního projektu tedy proběhne zhodnocení obcí 3. fáze mapování a jejich následná kategorizace, na jejímž základě bude určena časová náročnost zpracování jednotlivých obcí.

2 TVORBA VRSTEV

2.1 Přehled vrstev

Pro účely analýzy obcí byly z dostupných dat v databázi `pgis_uzpd` vytvořeny následující tematické vrstvy:

- obce (*ruian*)
- komunikace (*osm*)
- budovy (*osm*)

Tyto vrstvy pak byly doplněny vlastními vrstvami, které vymezují mapované obce 3. fáze (*obce_faze3_obyv*) a definují místa výjezdu jednotlivých mapovačů (*obce_vyjezdy*).

2.2 Postup vytvoření

2.2.1 Vlastní vrstvy

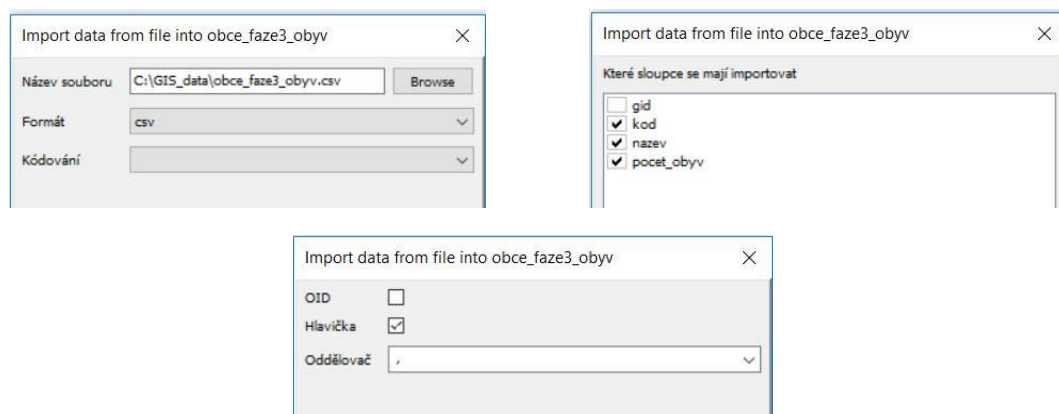
Nejprve byly do schématu importovány vlastní vrstvy *obce_faze3_obyv* a *obce_vyjezdy*. Data byla původně uložena ve formátu `.csv`. Nejprve byly ve schématu vytvořeny tabulky s předem definovaným primárním klíčem a sloupci, které odpovídaly struktuře uložení dat.

```
CREATE TABLE obce_faze3_obyv(gid serial PRIMARY KEY, kod INT, nazev VARCHAR(50),  
pocet_obyv INT);  
  
CREATE TABLE obce_vyjezdy(gid serial PRIMARY KEY, nazev VARCHAR(50), x_coord  
FLOAT, y_coord FLOAT);
```

Poté proběhl (neúspěšný) pokus importovat obě vrstvy pomocí příkazu `COPY`

```
COPY obce_faze3_obyv(kod, nazev, pocet_obyv) FROM  
'C:\GIS_data\obce_faze3_obyv.csv' WITH DELIMITER AS ',' CSV HEADER;
```

Dotaz však nemohl být vykonán, jelikož vyžadoval práva dostupná pouze pro roli superuživatele. Obě vrstvy tak byly nakonec importovány s využitím softwaru `pgAdmin III v1.22` (viz Obr. 1). Tabulka *obce_faze3_obyv* obsahoval pouze atributová data, nebylo tedy nutné ji dále zpracovávat. S ohledem na prostorový charakter dat tabulky *obce_vyjezdy* byla k tabulce přidán sloupec s geometrií v souřadnicovém systému WGS 84 Web Mercator (SRID 3857) a geometrie byla převzata ze souřadnic `x` a `y` uvedených v tabulce. Souřadnicový systém WGS 84 Web Mercator byl zvolen proto, že odpovídá souřadnicovému systému, ve kterém jsou uložena data z OSM, s kterými budu data kombinovat.



Obr. 1: Import dat v pgAdmin III

Jelikož data byla původně uložena v souřadnicovém systému S-JTSK Krovak East North (SRID 5514) bylo nutné během přenosu geometrie původní data transformovat.

```
ALTER TABLE obce_vyjezdy ADD COLUMN geom geometry(point, 3857);
UPDATE obce_vyjezdy
SET geom=st_transform(st_setsrid(st_point(x_coord, y_coord), 5514),3857);
```

Nakonec byl vytvořen prostorový index

```
CREATE INDEX obce_vyjezdy_gix ON obce_vyjezdy USING GIST (geom);
```

2.2.2 Obce 3. fáze

V dalším kroku byla propojením polygonové vrstvy obce ze schématu *ruian* a vrstvy *obce_faze3_obyv* vytvořena tabulka *obce_faze3*, která obsahuje data o 322 obcích, které budou mapovány v rámci terénního šetření

```
CREATE TABLE obce_faze3 AS
SELECT o.kod, o.nazev, o.geom, of3.pocet_obyv FROM obce AS o JOIN obce_faze3_obyv
AS of3 ON o.kod=of3.kod;
```

Data ze schématu *ruian* jsou uložena v souřadnicovém systému S-JTSK Krovak East North (SRID 5514), pro další práci s daty z OSM bylo nutné opět provést transformaci do souřadnicového systému WGS 84 Web Mercator (SRID 3857).

```
ALTER TABLE obce_faze3 ADD COLUMN geom1 geometry(multipolygon, 3857);
UPDATE obce_faze3 SET geom1 = ST_Transform(geom, 3857);
```

Poté byl nastaven primární klíč a vytvořen prostorový index

```
ALTER TABLE obce_faze3 ADD PRIMARY KEY (kod);
CREATE INDEX obce_faze3_gix1 ON obce_faze3 USING GIST (geom1);
```

2.2.3 Komunikace a budovy

Z dat OSM byly postupně vytvořeny vrstvy budov a komunikací. Nejprve byla výběrem příslušných prvků z tabulky *osm_line* ze schématu *osm* vytvořena tabulka komunikací. Kromě názvu sloupce (*highway*) určující kategorii prvků byl vybrán pouze sloupec *geom*. Na základě prostorového průniku s vrstvou *obce_faze3* pak byly vybrány komunikace pouze pro sledované obce

```
CREATE TABLE komunikace_faze3 AS
SELECT k.highway AS typ_kom, ST_Length(k.geom) AS delka_kom, k.geom, o.kod
FROM osm_line AS k
JOIN obce_faze3 AS o
ON st_intersects(k.geom, o.geom1) AND k.highway IN ('motorway', 'trunk',
'primary', 'secondary', 'tertiary', 'unclassified', 'residential', 'service',
'motorway_link', 'trunk_link', 'primary_link', 'secondary_link',
'tertiary_link', 'living_street', 'pedestrian', 'road');
```

Pro účely analýzy byly kromě dálnic a silnic 1., 2. a 3. třídy, vybrány také neklasifikované cesty, a komunikace v zástavbě včetně neprůjezdných ulic.

Z tabulky *osm_polygon* schématu *osm* pak byla podobným způsobem vytvořena tabulka budov.

Z tabulky *osm_polygon* byly převzaty všechny prvky ze sloupce *building* s vyplněnou hodnotou atributu (prvky s hodnotou NULL nebyly zahrnuty).

```
CREATE TABLE budovy_faze3 AS
SELECT b.building AS typ_budovy, b.geom, o.kod
FROM osm_polygon AS b JOIN obce_faze3 AS o ON st_intersects(b.geom, o.geom1)
WHERE b.building IS NOT NULL;
```

U obou tabulek byl následně nastaven primární klíč a vytvořen prostorový index

```
ALTER TABLE komunikace_faze3 ADD COLUMN gid serial;
ALTER TABLE komunikace_faze3 ADD PRIMARY KEY (gid);

ALTER TABLE budovy_faze3 ADD COLUMN gid serial;
ALTER TABLE budovy_faze3 ADD PRIMARY KEY (gid);

CREATE INDEX komunikace_faze3_gix ON komunikace_faze3 USING GIST (geom);

CREATE INDEX budovy_faze3_gix ON budovy_faze3 USING GIST (geom);
```

2.2.4 Zastavěné části

Z vrstvy budov vytvořené v předchozím kroku byla odvozena vrstva zastavěných částí, která vymezuje zájmový prostor pro terénní šetření. Zastavěná část je pro účely mapování definována jako shluk alespoň 5 budov, které od sebe nejsou vzdáleny více než 200 metrů. Na základě těchto kritérií byly zastavěné části vymezeny pomocí funkce *ST_ClusterDBSCAN*. Tato funkce přiřadí polygonům ve stejném shluku stejné CID neboli číslo shluku. S využitím těchto CID byly postupnou kombinací funkcí *ST_Collect*, *ST_ConcaveHull* a *ST_SimplifyPreserveTopology* vytvořeny polygony reprezentující zastavěné části.

```

CREATE TABLE zastavba_faze3 AS
(
WITH c AS (SELECT st_clusterDBSCAN(geom, 200, 5) over () as cid, b.geom
FROM budovy_faze3 AS b
)
SELECT cid,
st_simplifypreservetopology((st_concavehull(st_collect(c.geom),0.99)),1) AS
z_geom
FROM c
WHERE cid IS NOT NULL
GROUP BY cid
);

```

Následně byl opět nastaven primární klíč a vytvořen prostorový index

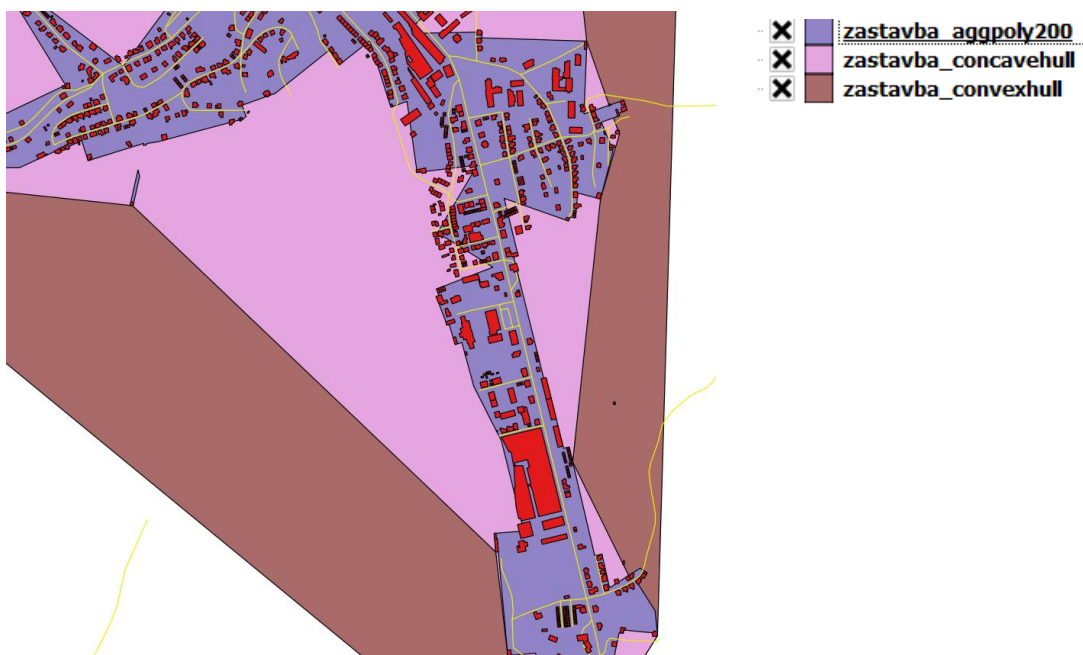
```

ALTER TABLE zastavba_faze3 ADD COLUMN gid serial;
ALTER TABLE zastavba_faze3 ADD PRIMARY KEY (gid);

CREATE INDEX zastavba_faze3_gix1 ON zastavba_faze3 USING GIST (z_geom);

```

ST_Collect sloučila geometrie budov uvnitř jednoho shluku do jedné geometrie. Každý shluk budov, který byl nyní reprezentován pouze jednou geometrií, pak byl pomocí funkce *ST_ConcaveHull* agregován do jednoho polygonu. Parametry funkce byly zvoleny s ohledem na výsledek a čas vykonání dotazu. Snížení hodnoty o 0.01 zpomaluje provedené dotazu kvadraticky [1]. Funkce *ST_SimplifyPreserveTopology* byl využita z důvodu zamezení vytvoření nevalidních geometrií na výstupu *ST_ConcaveHull* [2]. Funkce *ST_ConcaveHull* byla zvolena po dlouhém zkoušení jiných dostupných funkcí (např. *ST_ConvexHull* nebo autorské funkce *aggregatepolygons* [3]). Ze všech vyzkoušených funkcí poskytovala nejlepší výsledek. Výsledky (ne)použitých funkcí jsou porovnány na Obr. 2.



Obr. 2 Porovnání funkcí pro agregaci polygonů

2.3 Validace vrstev

Po vytvoření všech vrstev byly všechny vrstvy validovány s využitím funkce *ST_IsValid*. Žádná z vrstev neobsahoval nevalidní geometrii.

```
SELECT kod, ST_IsValidReason(geom1) AS duvod FROM obce_faze3  
WHERE ST_IsValid(geom1) = FALSE;
```

```
SELECT gid, ST_IsValidReason(geom) AS duvod FROM komunikace_faze3  
WHERE ST_IsValid(geom) = FALSE;
```

```
SELECT gid, ST_IsValidReason(geom) AS duvod FROM budovy_faze3  
WHERE ST_IsValid(geom) = FALSE;
```

```
SELECT gid, ST_IsValidReason(z_geom) AS duvod FROM zastavba_faze3  
WHERE ST_IsValid(z_geom) = FALSE;
```

```
SELECT gid, ST_IsValidReason(geom) AS duvod FROM obce_vyjezdy  
WHERE ST_IsValid(geom) = FALSE;
```


3 ANALÝZA

V analytické části jsou postupně pro každou obec 3. fáze vypočteny následující údaje

- počet zastavěných částí,
- délka komunikací v zastavěných částech,
- průměrná vzdálenost zastavěných částí v katastru obce,

Na základě těchto je pak provedena jednoduchá kategorizace obcí, které určuje časovou náročnost zpracování obce.

3.1 Výpočet ukazatelů

3.1.1 Počet zastavěných částí v obci

Pro zjištění informace o počtu zastavěných částí v jednotlivých obcích byla vytvořena pomocná tabulka *zastavba_faze3_obce*

```
CREATE TABLE zastavba_faze3_obce AS
SELECT o.kod as kod_obec, o.nazev as nazev_obec, z.cid as cid,
st_area(st_intersection(z.z_geom, o.geom1)) as zp_plocha,
st_intersection(z.z_geom, o.geom1) as zo_geom
FROM zastavba_faze3 AS z JOIN obce_faze3 AS o ON st_intersects(z.z_geom,
o.geom1)

ALTER TABLE zastavba_faze3_obce ADD COLUMN gid serial;

ALTER TABLE zastavba_faze3_obce ADD PRIMARY KEY (gid);

CREATE INDEX zastavba_faze3_obce_gix ON zastavba_faze3_obce USING GIST
(zo_geom);
```

Na základě této tabulky pak byl dopočten počet zastavěných částí v jednotlivých obcích, který byl přidán jako nový atribut *zp_pocet* do tabulky *obce_faze3*

```
ALTER TABLE obce_faze3
ADD COLUMN zp_pocet INT;

UPDATE obce_faze3
SET zp_pocet=z.zp_pocet
FROM
(
SELECT COUNT(cid) as zp_pocet, kod_obec
FROM zastavba_faze3_obce
GROUP BY kod_obec
) as z
WHERE obce_faze3.kod=z.kod_obec;
```

3.1.2 Průměrná vzdálenost zastavěných částí

Nejprve byla s využitím pomocné tabulky *zastavba_faze3_vzd* zjištěna vzájemná vzdálenost mezi zastavěnými částmi v rámci každé obce

```
CREATE TABLE zastavba_faze3_vzd AS
SELECT st_distance(st_centroid(z.zo_geom), st_centroid(zt.zo_geom)) as vzd,
z.cid as cid1, zt.cid as cid2, z.kod_obec as kod1, zt.kod_obec as kod2
FROM zastavba_faze3_obce as z, zastavba_faze3_obce as zt
WHERE z.kod_obec=zt.kod_obec AND z.cid<>zt.cid;
```

```
ALTER TABLE zastavba_faze3_vzd ADD COLUMN gid serial;
```

```
ALTER TABLE zastavba_faze3_vzd ADD PRIMARY KEY (gid);
```

Z tabulky musela být odstraněna duplicitní data, jelikož se vzdálenost mezi dvěma určitými zastavěnými částmi vyskytovala v tabulce vždy dvakrát. Duplicity byly odstraněny následujícím způsobem

```
DELETE
FROM zastavba_faze3_vzd as z1
USING zastavba_faze3_vzd as z2
WHERE z1.gid<z2.gid
AND z1.vzd=z2.vzd;
```

Získané údaje o vzdálenostech pak byly přepočteny na průměrnou vzdálenost pro každou obec a přidány jako nový atribut *zp_avg_vzd_km* do tabulky *obce_faze3*

```
ALTER TABLE obce_faze3
ADD COLUMN zp_avg_vzd_km FLOAT;

UPDATE obce_faze3
SET zp_avg_vzd_km=z.zp_avg_vzd_km
FROM
(
SELECT AVG(vzd)/1e3 as zp_avg_vzd_km, kod1
FROM zastavba_faze3_vzd
GROUP BY kod1
) as z
WHERE obce_faze3.kod=z.kod1;
```

3.1.3 Délka komunikací v zastavěné části obce

Celková délka komunikací v zastavěné části obce a nejbližším okolí 200 metrů byla dopočtena na základě prostorového propojení tabulek *zastavba_faze3_obce* a *komunikace_faze3*. Získaná data byla přímo nahrána do tabulky *obce_faze3* jako nový atribut *kom_delka_km*

```
ALTER TABLE obce_faze3
ADD COLUMN kom_delka_km FLOAT;
```

```

UPDATE obce_faze3
SET kom_delka_km=zk.kom_delka
FROM
(
WITH zb as
(
SELECT cid, kod_obec, st_buffer(z.zo_geom, 200) as geomb FROM
zastavba_faze3_obce AS z
)
SELECT st_length(st_collect(st_intersection(k.geom, zb.geomb)))/1e3 as
kom_delka, zb.kod_obec as kod_obec
FROM komunikace_faze3 AS k JOIN zb ON st_intersects(k.geom, zb.geomb)
GROUP BY zb.kod_obec
) as zk
WHERE obce_faze3.kod=zk.kod_obec;

```

3.2 Kategorizace obcí

Na základě každého z ukazatelů byly obce rozděleny do 3 kategorií, kdy kategorie 1 reprezentuje obce, které vykazují nejnižší hodnoty daného ukazatele; v kategorii 2 jsou obce, které mají průměrnou hodnotu daného ukazatele a do kategorie 3 jsou zařazeny obce, u nichž daný ukazatel nabývá nejvyšších hodnot. Intervaly pro jednotlivé ukazatele jsou uvedeny v Tab. 1.

	kategorie			
ukazatel	kat1	kat2	kat3	váha
počet obyvatel	3000–4999	5000–7499	7500–10000	0,1
délka komunikací	0–49	50–99	100+	0,2
počet zastavěných částí	1–9	10–19	20+	0,3
prům. vzdálenost zastavěných částí	0–2,99	3–5,99	6+	0,4

Tab. 1 Intervaly kategorií a váha ukazatelů

Do tabulky *obce_faze3* tedy byly přidány nové sloupce *zp_pocet_kat*, *zp_avg_vzd_km_kat*, *kom_delka_km_kat*, *pocet_obyv_kat*, do kterých byly uloženy hodnoty od 1 do 3 dle hodnoty příslušného ukazatele

```

ALTER TABLE obce_faze3
ADD COLUMN zp_pocet_kat INT,
ADD COLUMN zp_avg_vzd_km_kat INT,
ADD COLUMN kom_delka_km_kat INT,
ADD COLUMN pocet_obyv_kat INT;

UPDATE obce_faze3 SET zp_pocet_kat=1 WHERE zp_pocet < 10;
UPDATE obce_faze3 SET zp_pocet_kat=2 WHERE zp_pocet >= 10 AND zp_pocet < 20;
UPDATE obce_faze3 SET zp_pocet_kat=3 WHERE zp_pocet >= 20;

UPDATE obce_faze3 SET zp_avg_vzd_km_kat=1 WHERE zp_avg_vzd_km < 3 OR
zp_avg_vzd_km IS NULL;
UPDATE obce_faze3 SET zp_avg_vzd_km_kat=2 WHERE zp_avg_vzd_km >= 3 AND
zp_avg_vzd_km < 6;
UPDATE obce_faze3 SET zp_avg_vzd_km_kat=3 WHERE zp_avg_vzd_km >= 6;

```

```

UPDATE obce_faze3 SET kom_delka_km_kat=1 WHERE kom_delka_km < 50;
UPDATE obce_faze3 SET kom_delka_km_kat=2 WHERE kom_delka_km >= 50 AND
kom_delka_km < 100;
UPDATE obce_faze3 SET kom_delka_km_kat=3 WHERE kom_delka_km >= 100;

UPDATE obce_faze3 SET pocet_obyv_kat=1 WHERE pocet_obyv < 5000;
UPDATE obce_faze3 SET pocet_obyv_kat=2 WHERE pocet_obyv >= 5000 AND pocet_obyv
< 7500;
UPDATE obce_faze3 SET pocet_obyv_kat=3 WHERE pocet_obyv >= 7500;

```

Na závěr byla každé obci přidělena celková kategorie náročnosti, která byla vypočtena jako vážený průměr výše uvedených kategorií ukazatelů. Každému z ukazatelů byla přidělena různá váha dle toho, jakým způsobem se promítají do náročnosti zpracování obce. Na základě zkušeností mapovačů se časová náročnost obce odvíjí především od počtu zastavěných částí a jejich vzdálenosti v rámci katastru obce. Těmto dvou ukazatelům proto byla přidělena nejvyšší váha (viz Tab. 1). Na základě celkové hodnoty kategorie pak byla každé obci přidělena hodnota časové náročnosti jejich zpracování, a to 8 hodin (kategorie 1), 12 hodin (kategorie 2) a 16 hodin (kategorie 3). Časová náročnost byla opět určena na základě zkušeností z předchozích mapování. Do tabulky *obce_faze3* tedy byly přidány poslední dva sloupce *kat_celkem* a *zpracovani_hod*, do kterých byly nahrány příslušné hodnoty

```

ALTER TABLE obce_faze3
ADD COLUMN kat_celkem INT;
ADD COLUMN zpracovani_hod INT;

UPDATE obce_faze3 SET
kat_celkem=((0.3*zp_pocet_kat)+(0.4*zp_avg_vzd_km_kat)+(0.2*kom_delka_km_kat)+
(0.1*pocet_obyv_kat));

UPDATE obce_faze3 SET zpracovani_hod=8 WHERE kat_celkem=1;
UPDATE obce_faze3 SET zpracovani_hod=12 WHERE kat_celkem=2;
UPDATE obce_faze3 SET zpracovani_hod=16 WHERE kat_celkem=3;

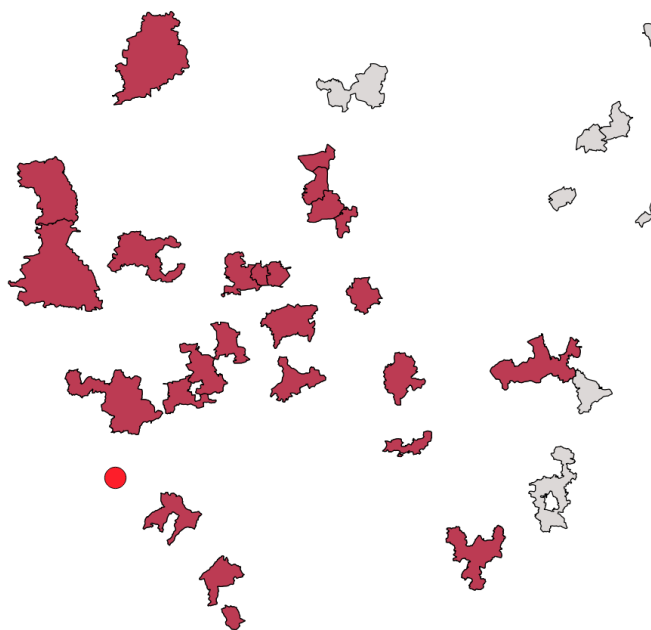
```

4 PROSTOROVÉ A ATRIBUTOVÉ DOTAZY

1. Kolik mapovaných obcí leží v okruhu 100 km od Domažlic a jaká je jejich časová náročnost zpracování přepočtena na počet pracovních dnů (pracovní den = 8 hodin)?

```
SELECT COUNT (o.kod) AS obce_pocet, SUM(zpracovani_hod)/8 AS zpracovani_dny
FROM obce_faze3 AS o
JOIN obce_vyjezdy AS v
ON ST_DWithin(o.geom1, v.geom, 1e5)
WHERE v.nazev = 'Domažlice'
```

	obce_pocet	zpracovani_dny
1	24	35



2. Jaký je počet obcí v jednotlivých kategoriích?

```
SELECT COUNT(kod), kat_celkem
FROM obce_faze3
GROUP BY kat_celkem;
```

	count	kat_celkem
1	84	1
2	188	2
3	50	3

3. Jaký počet kilometrů komunikací musí projít mapovač z Prahy-Smíchova, který bude mapovat obce v okruhu 50 km? Jak dlouho by mu trvalo tyto komunikace projít, pokud budeme uvažovat průměrnou rychlost chůze 4 km/hod.?

```
SELECT SUM(o.kom_delka_km) as pocet_km, (SUM(o.kom_delka_km))/4 as pocet_hodin,
((SUM(o.kom_delka_km))/4)/8 as pocet_dni
FROM obce_faze3 AS o
JOIN obce_vyjezdy AS v
ON ST_DWithin(o.geom1, v.geom, 50000)
WHERE v.nazev = 'Praha-Smíchov';
```

	pocet_km	pocet_hodin	pocet_dni
1	2512.3640607	628.091015175	78.5113768969

4. Pro porovnání: Jaká je předem stanovená časová náročnost zpracování obcí z předešlého dotazu?

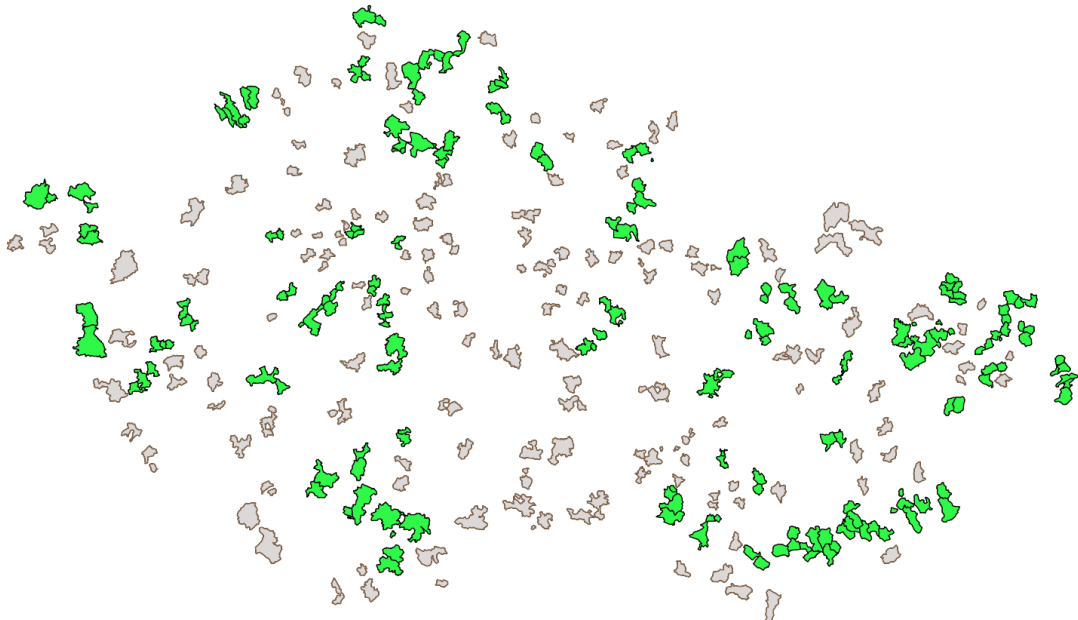
```
SELECT COUNT (o.kod) AS obce_pocet, SUM(zpracovani_hod)/8 AS zpracovani_dny
FROM obce_faze3 AS o
JOIN obce_vyjezdy AS v
ON ST_DWithin(o.geom1, v.geom, 50000)
WHERE v.nazev = 'Praha-Smíchov';
```

	obce_pocet	zpracovani_dny
1	34	43

5. Kolik mapovaných obcí spolu sousedí?

```
SELECT COUNT(DISTINCT o.kod)
FROM obce_faze3 as o, obce_faze3 as o1
WHERE st_touches(o.geom1, o1.geom1)=true;
```

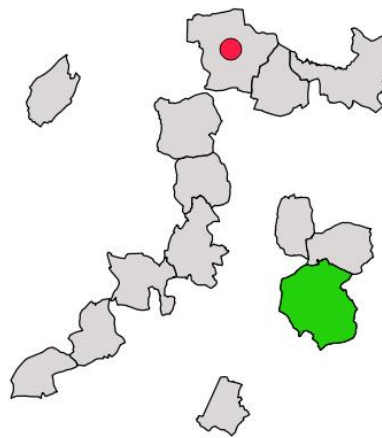
	count
1	172



6. V jaké obci bude muset mapovač navštívit nejvíce zastavěných částí - mapovač z které obce to bude mít nejblíže (kolik km to bude)?

```
SELECT o.nazev, o.zp_pocet, v.nazev as obec_vyjezd, st_distance(o.geom1,
v.geom)/1e3 as vzd
FROM obce_faze3 as o, obce_vyjezdy as v
ORDER BY zp_pocet DESC, vzd ASC
LIMIT 1;
```

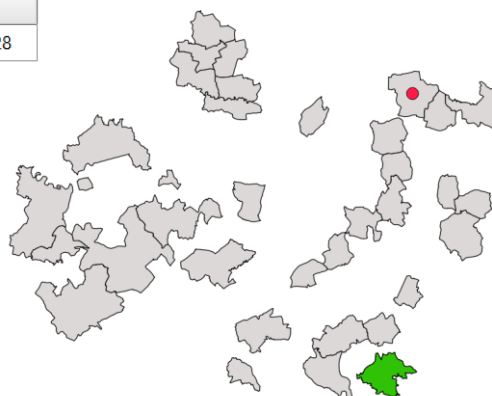
	nazev	zp_pocet	obec_vyjezd	vzd
1	Těrlicko	48	Dolní Lutyně	25.4463277...



7. V jaké obci bude muset mapovač zmapovat nejvíce kilometrů komunikací - mapovač z které obce to bude mít nejblíže (kolik km to bude)?

```
SELECT o.nazev, o.kom_delka_km, v.nazev as obec_vyjezd, st_distance(o.geom1,
v.geom)/1e3 as vzd
FROM obce_faze3 as o, obce_vyjezdy as v
ORDER BY kom_delka_km DESC, vzd ASC
LIMIT 1;
```

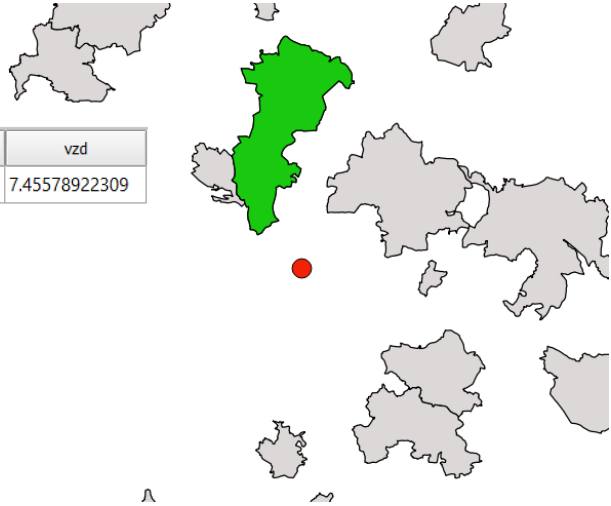
	nazev	kom_delka_km	obec_vyjezd	vzd
1	Frýdlant nad Ostravicí	185.00098633	Dolní Lutyně	52.0361550928



8. V jaké obci bude muset mapovač najezdit nejvíce kilometrů mezi zastavěnými částmi - mapovač z které obce to bude mít nejbližší (kolik km to bude)?

```
SELECT o.nazev, o.zp_avg_vzd_km, v.nazev as obec_vyjezd, st_distance(o.geom1,
v.geom)/1e3 as vzd
FROM obce_faze3 as o, obce_vyjezdy as v
WHERE zp_avg_vzd_km IS NOT NULL
ORDER BY zp_avg_vzd_km DESC, vzd ASC
LIMIT 1;
```

	nazev	zp_avg_vzd_km	obec_vyjezd	vzd
1	Hluboká nad Vltavou	10.2130660936	České Budějovice	7.45578922309



5 ZÁVĚR

V rámci semestrálního projektu byla provedena analýzy vybraných obcí ČR pro účely terénního šetření probíhajícího v rámci projektu „Mapování infrastruktury a služeb venkova ČR“. Na základě dat z OSM a RUIAN byly v kombinaci s vlastními vrstvami vymezujícími sledované obce vypočteny celkem 4 ukazatele, na jejichž základě byla provedena kategorizace obcí, podle které byla stanovena časová náročnost jejich zpracování. Na základě výsledných dat bude možné provést rozdělení mapovacích oblastí mezi jednotlivé mapovače a zavést spravedlivé finanční ohodnocení.

Použitý software

- QGIS v2.18.28
- pgAdmin III v1.22

Reference

[1] https://postgis.net/docs/ST_ConcaveHull.html

[2] https://postgis.net/docs/ST_SimplifyPreserveTopology.html

[3] <https://github.com/hdus/pgtools/blob/master/aggregatepolygons/aggregatePolygons.sql>

<https://github.com/hdus/pgtools/blob/master/cleangeometry/cleanGeometry.sql>