

# KATEGORIZACE OBCÍ PRO TERÉNNÍ ŠETŘENÍ OBCÍ ČR

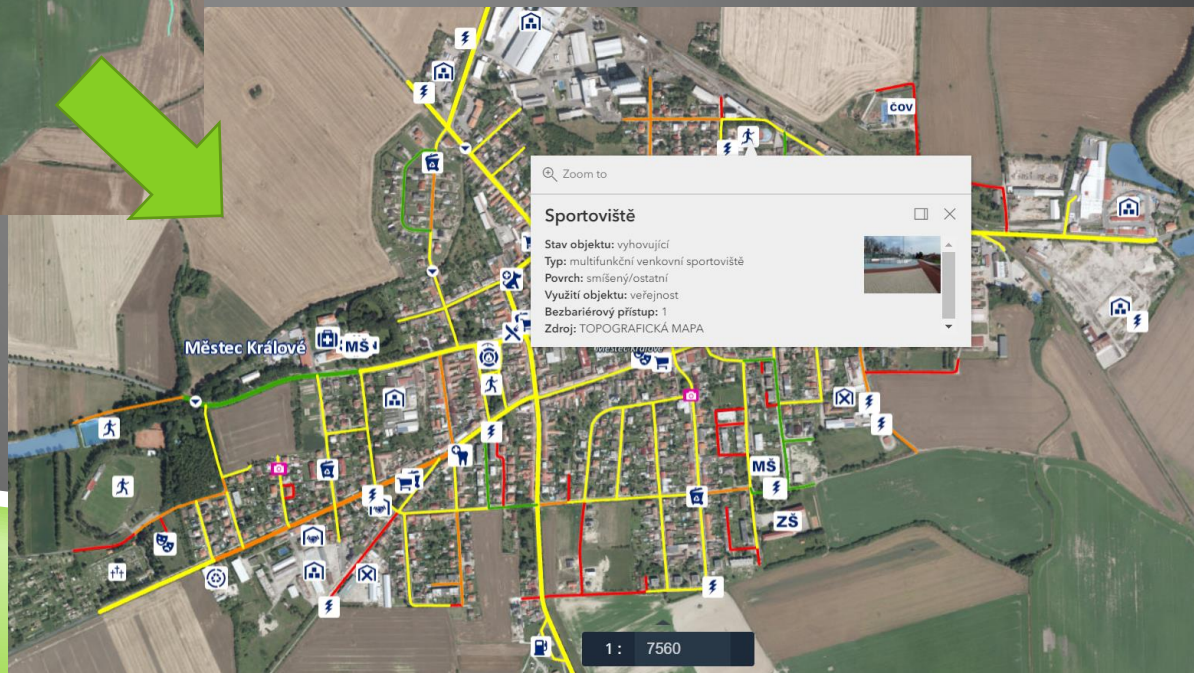
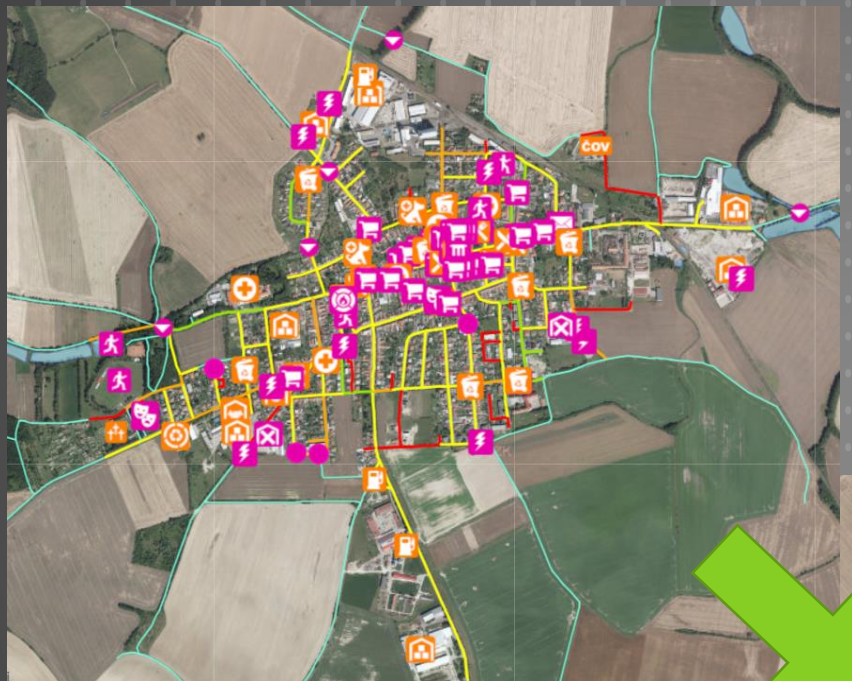
Petra Jílková  
FSv, ČVUT, katedra geomatiky

I55UZPD, ZS2018/2019

# ÚVOD

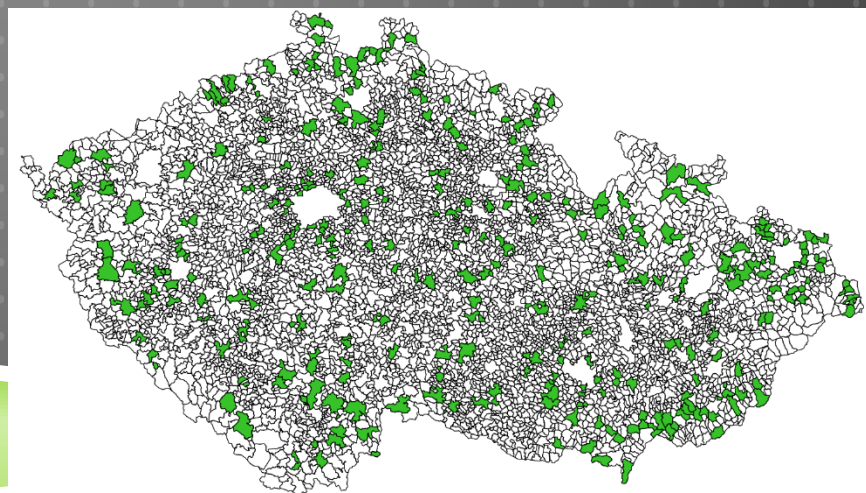
- ▶ Analýza obcí pro účely terénního šetření
- ▶ Projekt „Mapování infrastruktury a služeb venkova“
  - ▶ SMS ČR, MMR, MŠMT
  - ▶ Obce do 10 000 obyvatel
  - ▶ Aktuální stav komunikací a vybraných zájmových bodů (hodnocení+fotografie)
  - ▶ Terénní data+státní registry → AGIS SMS ČR

# ÚVOD



# 3. FÁZE MAPOVÁNÍ

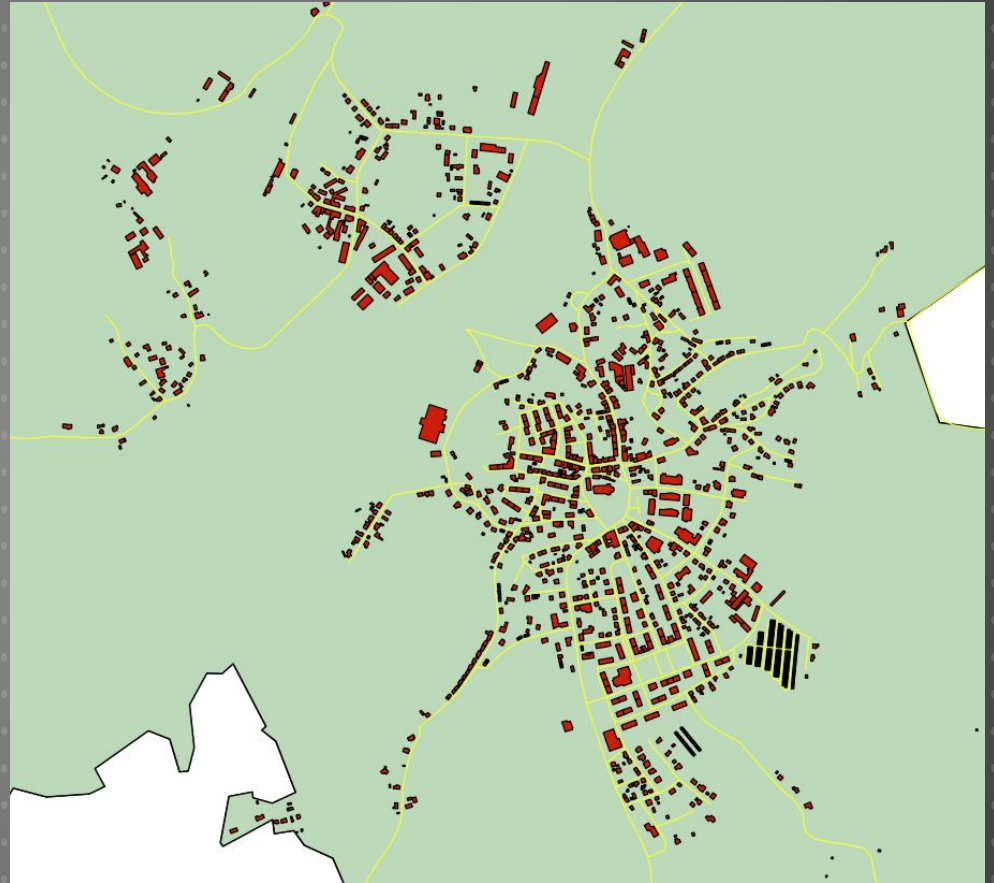
- ▶ Obce od 3 000 do 10 000 obyvatel
- ▶ Stanovení časové náročnosti zpracování obcí dle 4 kritérií
  - ▶ Počet obyvatel
  - ▶ Počet zastavěných částí
  - ▶ Délka komunikací v zastavěných částech
  - ▶ Průměrná vzdálenost zastavěných částí v katastru obce





# DATOVÉ VRSTVY

- ▶ OSM
  - ▶ komunikace (highways)
  - ▶ budovy (buildings)
- ▶ RUIAN
  - ▶ obce
- ▶ vlastní
  - ▶ obce\_faze3\_obyv
  - ▶ obce\_vyjezdy



# TVORBA VRSTEV

- ▶ vlastní vrstvy
- ▶ obce 3. fáze (RUIAN)
- ▶ budovy a komunikace (OSM)
- ▶ zastavěné části

# VLASTNÍ VRSTVY

- ▶ *obce\_faze3\_obyv, obce\_vyjezdy*

- ▶ založení tabulek

```
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);
```

- ▶ import dat z csv (příkaz COPY vs. pgAdmin)

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

# VLASTNÍ VRSTVY

## ► *obce\_vyjezdy*

## ► přenos geometrie+transformace dat (5514 → 3857)

```
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);
```

	gid	nazev	x_coord	y_coord	geom
1	3	Olomouc	-545557.406...	-1121431.41...	POINT
2	4	České Budě...	-755321.419...	-1165562.05...	POINT
3	5	Tuhaň	-733337.981...	-991750.3904	POINT
4	6	Pelhřimov	-694813.454...	-1124180.80...	POINT
5	7	Humpolec	-685194.739	-1112903.34...	POINT



# OBCE 3. FÁZE

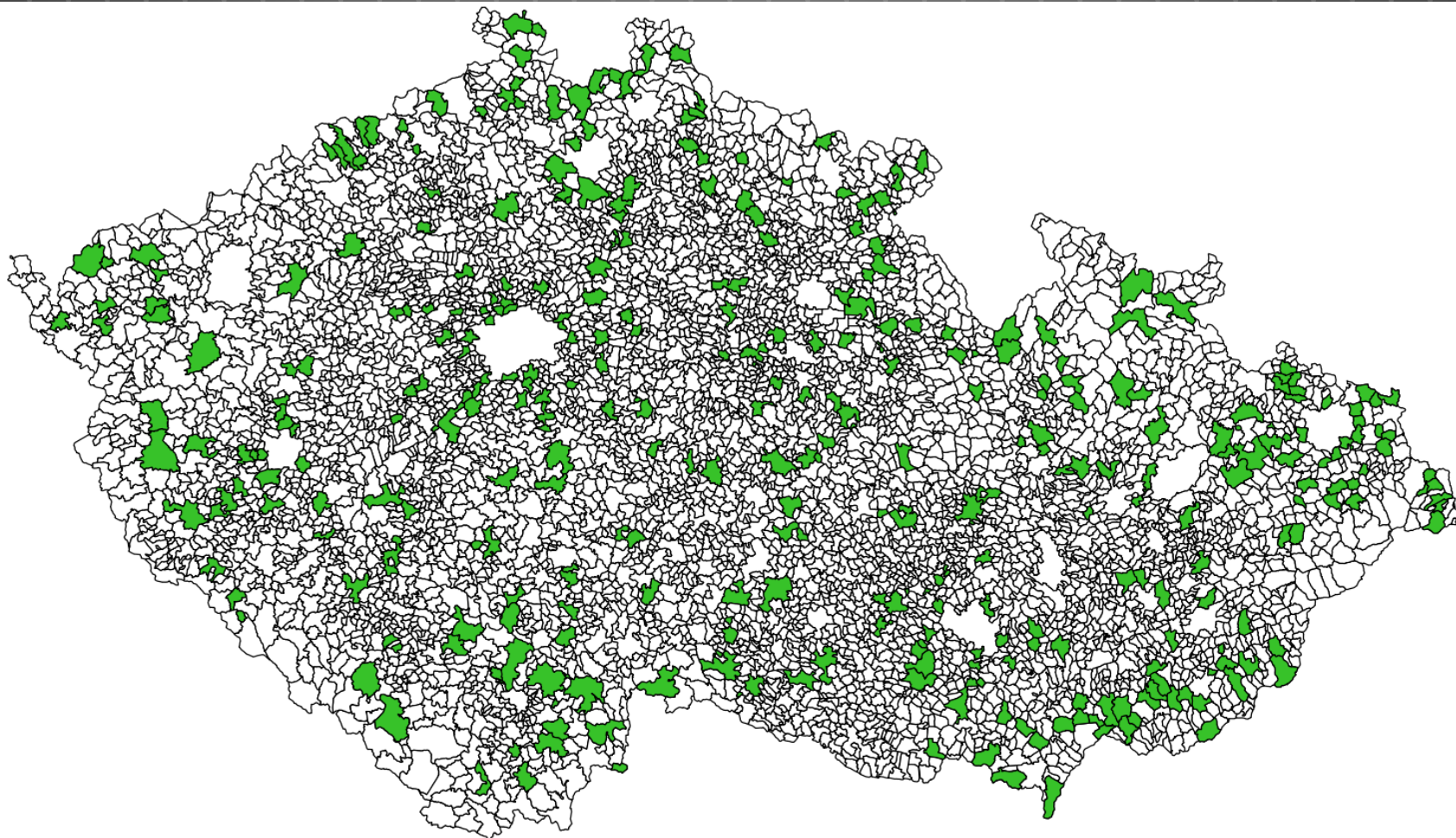
## ► *obce (RUIAN)+obce\_faze3\_obyv*

```
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;
```

## ► *transformace (5514 → 3857)*

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

# OBCE 3. FÁZE



# KOMUNIKACE A BUDOVY

## ► *komunikace\_faze3*

```
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');
```

## ► *prostorový průnik s obce\_faze3*

# KOMUNIKACE A BUDOVY

## ► *budovy\_faze3*

```
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;
```

## ► prostorový průnik s *obce\_faze3*

# ZASTAVĚNÉ ČÁSTI

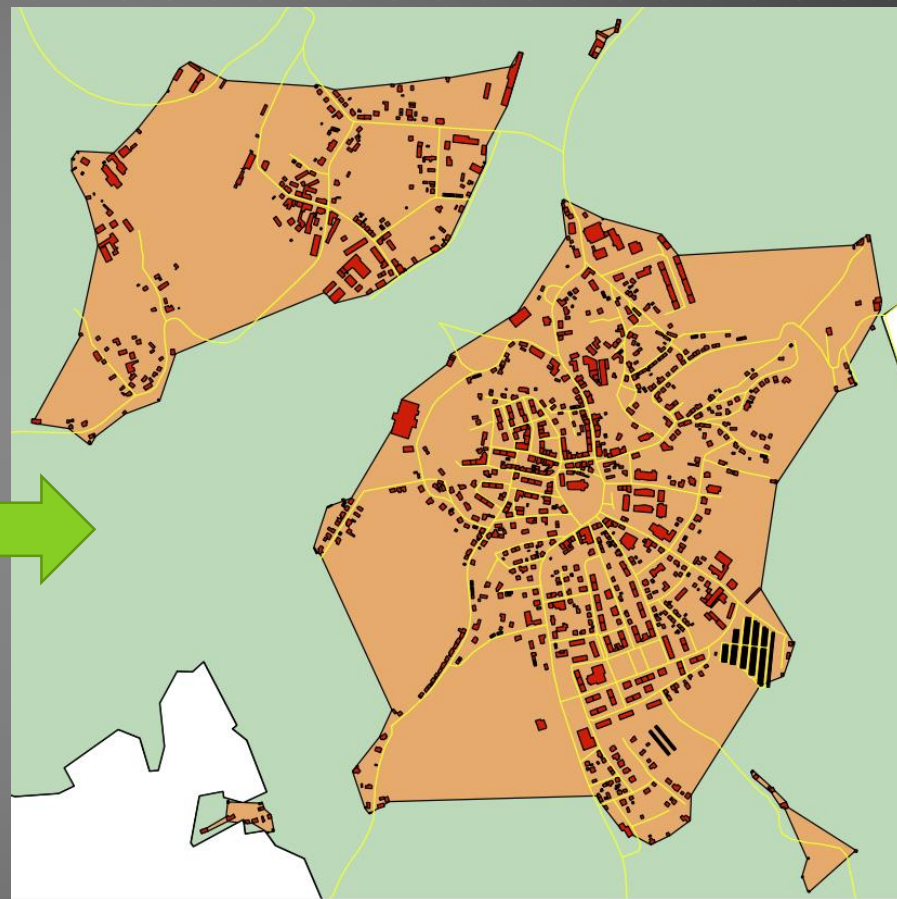
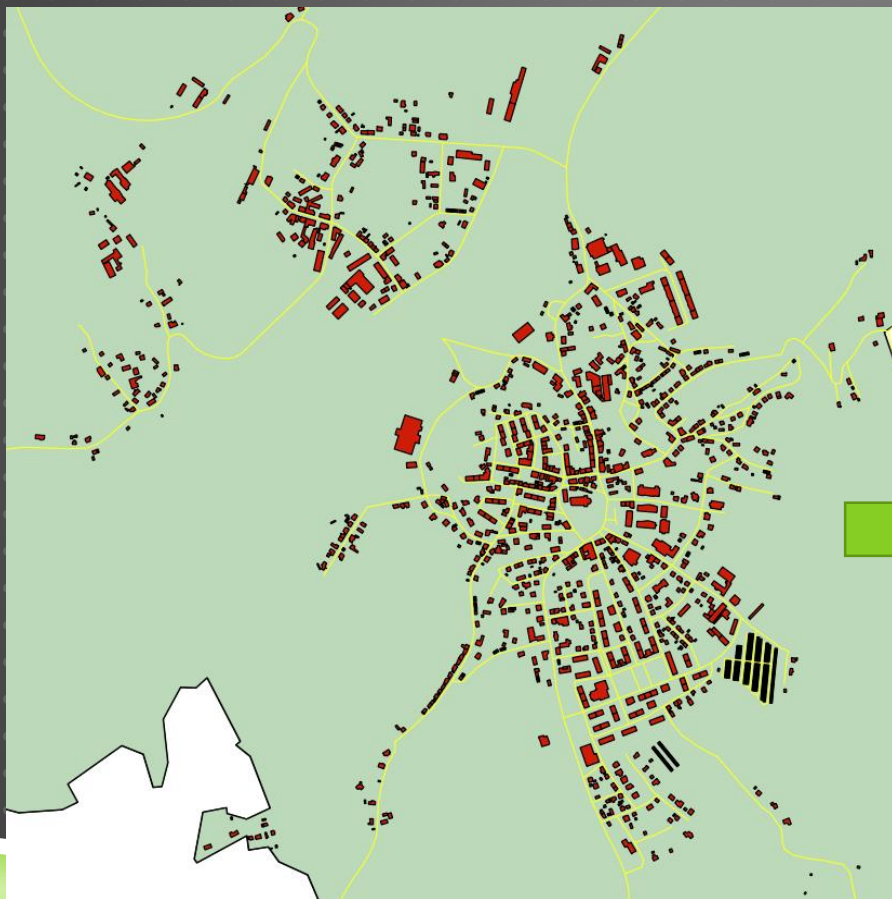
- ▶ *budovy\_faze3* → *zastavba\_faze3*
- ▶ shluk nejméně 5 budov vzdálených do 200 m

```
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);
```

▶ *ST\_ClusterDBSCAN* → *ST\_Collect* → *ST\_ConcaveHull*

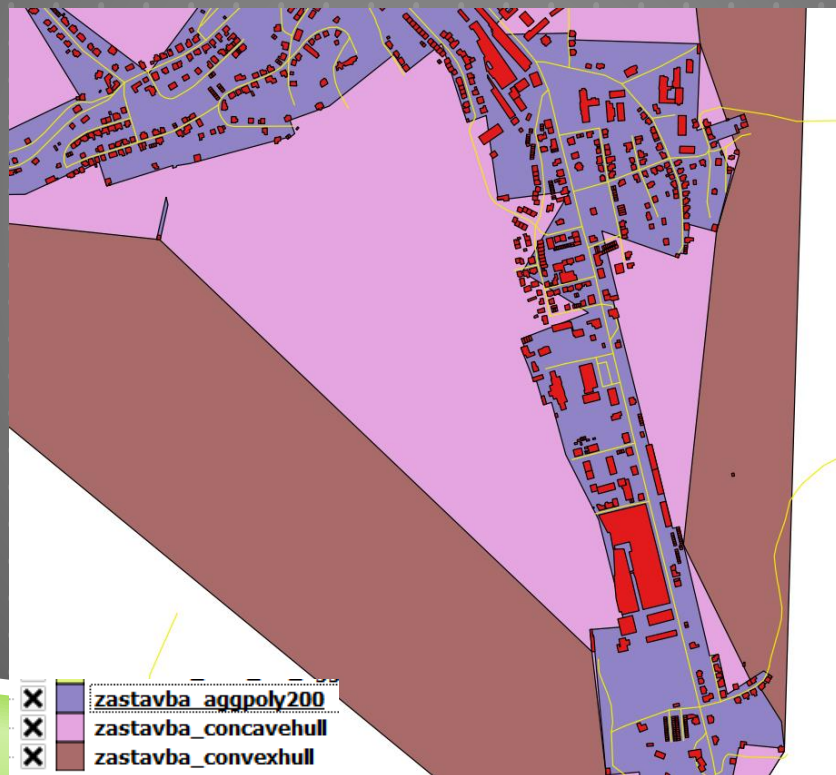


# ZASTAVĚNÉ ČÁSTI



# ZASTAVĚNÉ ČÁSTI

- ▶ různé postupy
- ▶ *ST\_ConvexHull* vs. *ST\_ConcaveHull* vs. *aggregatepolygons*



# VALIDACE

- ▶ *ST\_IsValid*

- ▶ všechny vrstvy OK

```
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;
```

...

# ANALÝZA OBCÍ 3. FÁZE

- ▶ Výpočet ukazatelů
  - ▶ Počet zastavěných částí
  - ▶ Průměrná vzdálenost
  - ▶ Délka komunikací
- ▶ Kategorizace obcí

	kod	nazev	geom	pocet_obyv	
1	510483	Štěpánkovice	SRID=5514...	3216	
2	586510	Ratíškovice	SRID=5514...	4028	
3	506214	Bolatice	SRID=5514...	4443	
4	559491	Tlučná	SRID=5514...	3244	
5	503657	Lutín	SRID=5514...	3197	

# POČET ZASTAVĚNÝCH ČÁSTÍ

► *zastavba\_faze3* x *obce\_faze3* → *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)
```

► atribut *zp\_pocet* → *obce\_faze3*

	kod	nazev	geom	pocet_obyv	zp_pocet
1	510483	Štěpánkovice	SRID=5514...	3216	4
2	586510	Ratíškovice	SRID=5514...	4028	6
3	506214	Bolatice	SRID=5514...	4443	2



# POČET ZASTAVĚNÝCH ČÁSTÍ

► atribut *zp\_pocet* → *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;
```

	kod	nazev	geom	pocet_obyv	zp_pocet
1	510483	Štěpánkovice	SRID=5514...	3216	4
2	586510	Ratíškovice	SRID=5514...	4028	6
3	506214	Bolatice	SRID=5514...	4443	2

# PRŮMĚRNÁ VZDÁLENOST ZASTAVĚNÝCH ČÁSTÍ

► *zastavba\_faze3\_obce x zastavba\_faze3\_obce → zastavba\_faze3\_obce\_vzd*

```
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;
```

► odstranění duplicit

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

# PRŮMĚRNÁ VZDÁLENOST ZASTAVĚNÝCH ČÁSTÍ

► atribut `zp_avg_vzd_km` → `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;
```

	kod	nazev	geom	pocet_obyv	zp_pocet	zp_avg_vzd_km
1	510483	Štěpánkovice	SRID=5514...	3216	4	4.1359860757...
2	586510	Ratíškovice	SRID=5514...	4028	6	2.8791919101...
3	506214	Bolatice	SRID=5514...	4443	2	2.9218760236...

# DÉLKA KOMUNIKACÍV ZÁSTAVBĚ

► *zastavba\_faze3\_obce* x *komunikace\_faze3* → 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
```

	kod	nazev	geom	pocet_obyv	zp_pocet	zp_avg_vzd_km	kom_delka_km
1	510483	Štěpánkovice	SRID=5514...	3216	4	4.1359860757...	34.324075680...
2	586510	Ratíškovice	SRID=5514...	4028	6	2.8791919101...	38.182861367...

```
(  
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;
```

# KATEGORIZACE OBCÍ

- rozdělení ukazatelů do kategorií

	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

- atributy *zp\_pocet\_kat*, *zp\_avg\_vzd\_km\_kat*, *kom\_delka\_km\_kat*, *pocet\_obyv\_kat* → *obce\_faze3*

kod	nazev	geom	pocet_obyv	zp_pocet	zp_avg_vzd_km	kom_delka_km	zp_pocet_kat	zp_avg_vzd_km_k	kom_delka_km_ki
510483	Štěpánkovice	SRID=5514...	3216	4	4.1359860757...	34.324075680...	1	2	1
586510	Ratíškovice	SRID=5514...	4028	6	2.8791919101...	38.182861367...	1	1	1
506214	Bolatice	SRID=5514...	4443	2	2.9218760236...	45.162268187...	1	1	1



# KATEGORIZACE OBCÍ

```
ALTER TABLE obce_faze3  
ADD COLUMN zp_pocet_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 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;
```

```
...
```

# KATEGORIZACE OBCÍ

- celková kategorie (váhy ukazatelů) → časová náročnost

```
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;
```

kod	nazev	geom	pocet_obyv	zp_pocet	zp_avg_vzd_km	kom_delka_km	zp_pocet_kat	zp_avg_vzd_km_kat	kom_delka_km_kat	pocet_obyv_kat	kat_celkem	zpracovani_hod
510483	Štěpánkovice	SRID=5514...	3216	4	4.1359860757...	34.324075680...	1	2	1	1	1	8
586510	Ratíškovice	SRID=5514...	4028	6	2.8791919101...	38.182861367...	1	1	1	1	1	8
506214	Bolatice	SRID=5514...	4443	2	2.9218760236...	45.162268187...	1	1	1	1	1	8

# PROSTOROVÉ 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)?

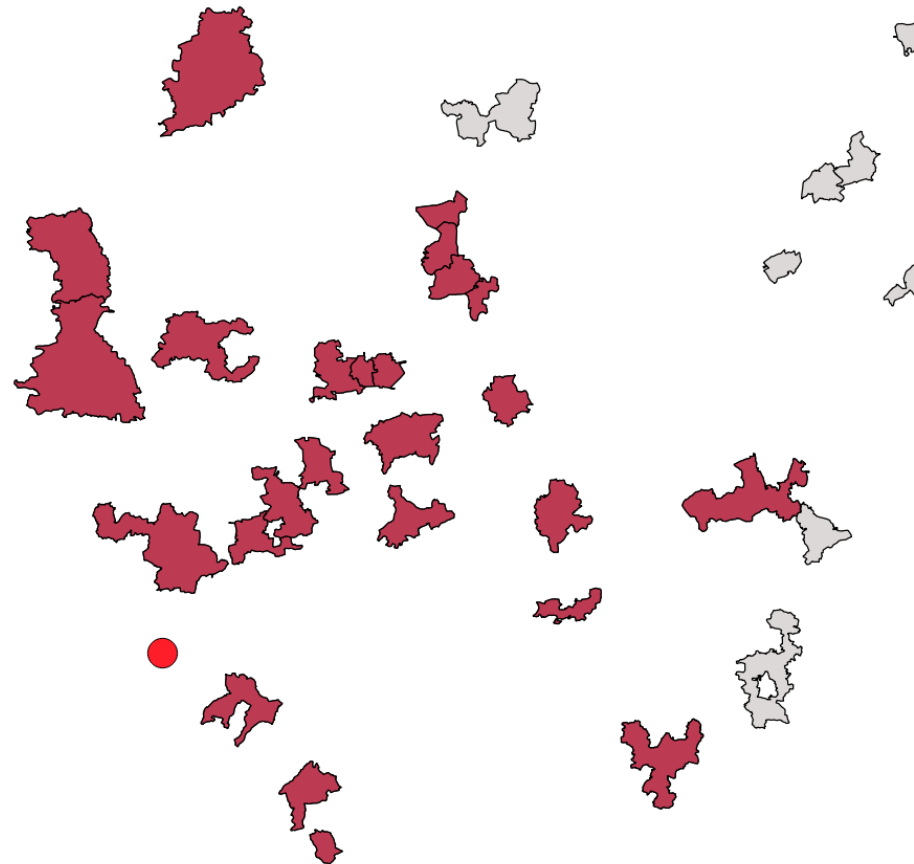
```
SELECT COUNT (o.kod) AS obce_pocet, SUM(zpracovani_hod)/8 AS zpracovani_dny  
FROM obce_faze3 AS o  
JOIN obce_vyjezdy AS v  
ON ST_DWithin(o.geom1, v.geom, 1e5)  
WHERE v.nazev = 'Domažlice'
```

# PROSTOROVÉ 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



# PROSTOROVÉ DOTAZY

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';
```



# PROSTOROVÉ DOTAZY

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

# PROSTOROVÉ DOTAZY

Kolik mapovaných obcí spolu sousedí?

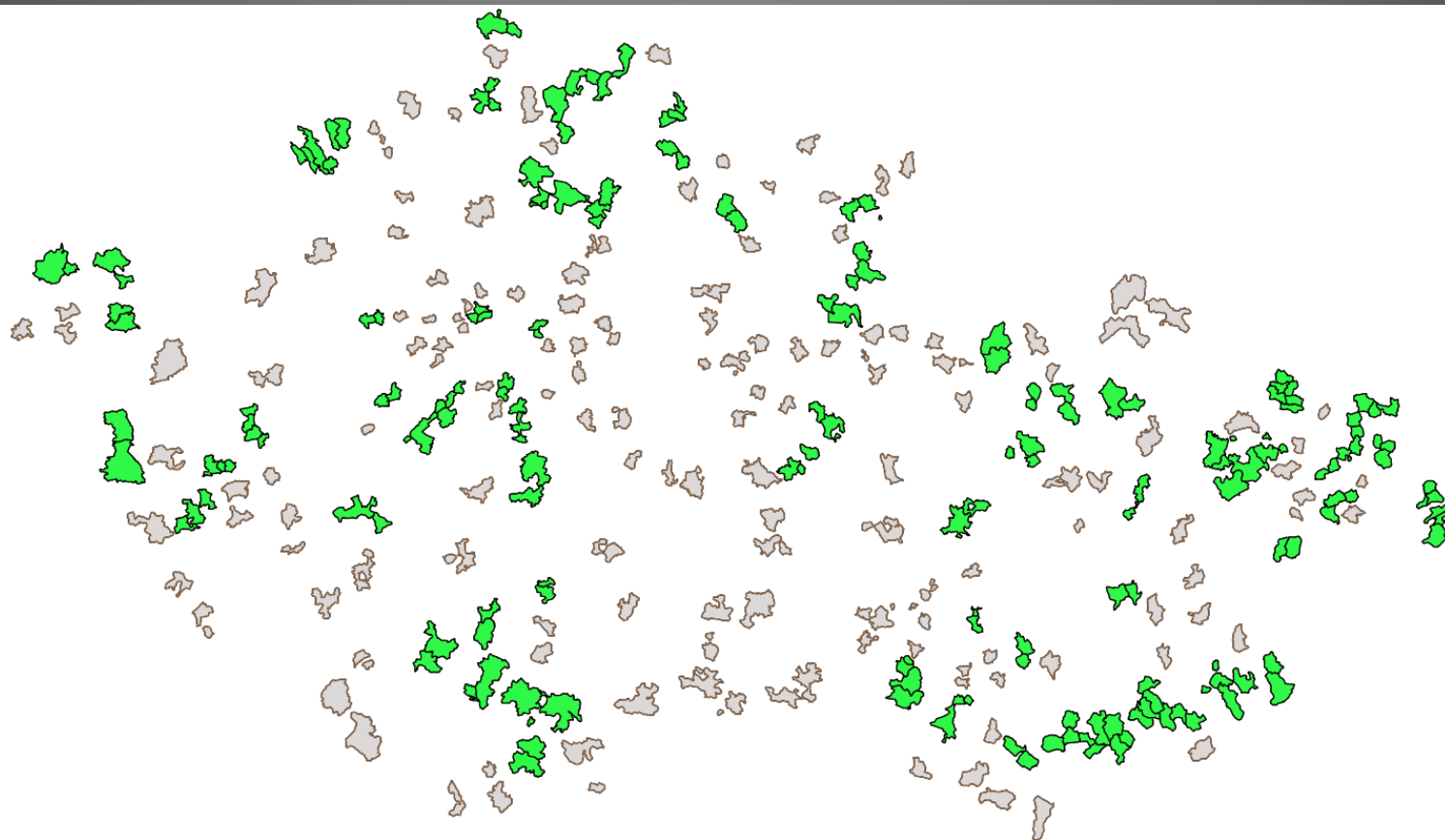
```
SELECT COUNT(DISTINCT o.kod)  
FROM obce_faze3 as o, obce_faze3 as o1  
WHERE st_touches(o.geom1, o1.geom1)=true;
```

# PROSTOROVÉ DOTAZY

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



# PROSTOROVÉ DOTAZY

V jaké obci bude muset mapovač navštívit nejvíce zastavěných částí - mapovač z které obce to bude mít nejbližší (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;
```

# PROSTOROVÉ DOTAZY

V jaké obci bude muset mapovač navštívit nejvíce zastavěných částí - mapovač z které obce to bude mít nejbližše (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...

