

# Semestrální projekt

## Úvod do zpracování prostorových dat

Václav Petráš  
Nikola Němcová  
Anna Kratochvílová

České vysoké učení technické v Praze  
Fakulta stavební

13. května 2010

- 1 Úvod
- 2 Validace lesů
  - Validace polygonů
  - Překryt polygonů
- 3 Ukázky dotazů
- 4 Shrnutí

## 1 Úvod

## 2 Validace lesů

- Validace polygonů
- Překryt polygonů

## 3 Ukázky dotazů

## 4 Shrnutí

- výběr tématických vrstev na základě dat OSM
- testy datové integrity
- sada dotazů nad tabulkami

- vybrané vrstvy
  - polygonová vrstva lesů (`landuse='forest'`)
  - polygonová vrstva vojenských újezdů  
(`landuse='danger area' AND name LIKE '%újezd%'`)
  - bodová vrstva bunkrů (`military='bunker'`)
    - `czech_point`
    - `czech_polygon` pomocí `ST_Centroid`

## Validita lesů

Z 67146 polygonů je 289 nevalidních.

## 1 Úvod

## 2 Validace lesů

- Validace polygonů
- Překryt polygonů

## 3 Ukázky dotazů

## 4 Shrnutí

- 1 Úvod
- 2 Validace lesů
  - Validace polygonů
  - Překryt polygonů
- 3 Ukázky dotazů
- 4 Shrnutí

Ke zjištění validity slouží funkce:

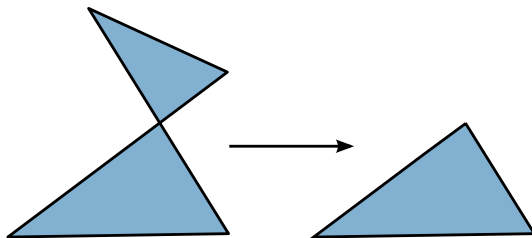
- `ST_IsValid`
- `ST_IsValidReason`

## Vyskytující se chyby

- Self-intersection
- Hole lies outside shell
- Ring Self-intersection
- Duplicate Rings
- Holes are nested



- funkce `ST_SnapToGrid(geometry A, float size)`
- především funkce `ST_Buffer(geometry A, 0.0)` pro chybu typu *self-intersection*



- inspirace funkcí cleanGeometry

```
SELECT DISTINCT
  gid,
  ST_ExteriorRing((ST_DumpRings(way)).geom) AS way
FROM f10.lesni_porosty
WHERE NOT ST_IsValid(way)
```

- inspirace funkcí cleanGeometry

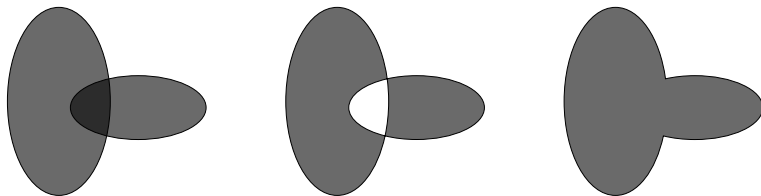
```
SELECT
  gid,
  ST_BuildArea(ST_Union(ring.way, ST_StartPoint(ring.way)))
  AS way
FROM
  (SELECT DISTINCT
    gid,
    ST_ExteriorRing((ST_DumpRings(way)).geom) AS way
  FROM f10.lesni_porosty
  WHERE NOT ST_IsValid(way))
AS ring
```

# Náš postup validace

- inspirace funkcí cleanGeometry

```
SELECT gid, ST_BuildArea(ST_Collect(way)) AS way
FROM
(SELECT
  gid,
  ST_BuildArea(ST_Union(ring.way, ST_StartPoint(ring.way)))
  AS way
FROM
(SELECT DISTINCT
  gid,
  ST_ExteriorRing((ST_DumpRings(way)).geom) AS way
FROM f10.lesni_porosty
WHERE NOT ST_IsValid(way))
AS ring)
AS after_union
GROUP BY gid;
```

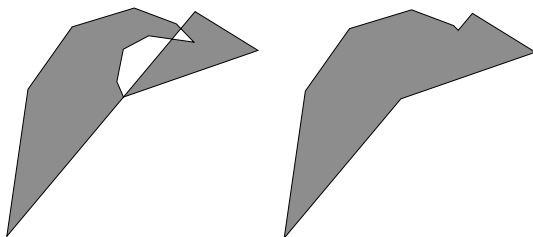
- funkce `ST_BuildArea` a `ST_Union`



Obrázek: vstup, výstup `ST_BuildArea`, výstup `ST_Union`

- ověření validity pomocí `ST_IsValid`
- ověření tvaru polygonů
  - porovnání výměr určených funkcí `ST_Area` – problematické
  - porovnání počtu ringů
    - zvětšení počtu – self-intersection
    - stejný počet – hole lies outside shell
    - menší počet – duplicate rings
  - kontrola vizuální (*QGis*) – v omezené míře

- nezjištěn žádný větší nesoulad
- nalezen atypický polygon se změněným tvarem



- otázka interpretace nevalidních polygonů v *QGisu*
- nejasný úmysl pořizovatele dat

## 1 Úvod

## 2 Validace lesů

- Validace polygonů
- Překryt polygonů

## 3 Ukázky dotazů

## 4 Shrnutí



- doposud řešeny problémy jen v rámci jednoho polygonu
- některé polygony se vzájemně překrývají
  - částečně (při okrajích)
  - většinou své plochy (duplicita)
- při řešení je možné vzít v úvahu
  - velikost plochy
  - hodnoty atributů
  - zdroj dat

## Závěr

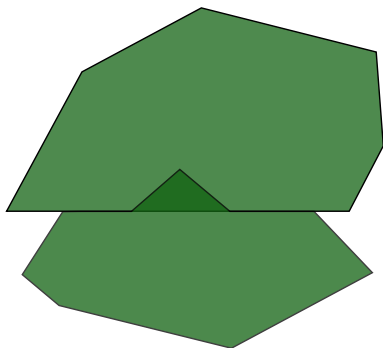
Vzhledem k různorodosti dat je těžké najít optimální řešení pro všechny polygony a toto řešení by bylo komplikované.

# Náš způsob řešení

- spojí se překryty patřící k jednomu polygonu
- pro každý polygon uvažován překryt s polygonem s vyšším id
- překryty se neopakují
- odečtení překrytu od původního polygonu – dle id

```
SELECT g1.gid
       ST_Union(ST_Buffer(ST_Intersection(g1.way, g2.way), 0.0))
FROM f10.lesni_porosty g1, f10.lesni_porosty g2
WHERE g1.gid < g2.gid
      AND ST_Intersects(g1.way, g2.way)
      AND NOT ST_Touches(g1.way, g2.way)
GROUP BY g1.gid
```

- funkce `ST_Intersection` může vrátit `GEOMETRYCOLLECTION`



- použití `ST_Buffer(geometry, 0.0)`

- funkce `ST_Union` vyvolala výjimku

```
NOTICE: TopologyException: found non-noded intersection  
between ...
```

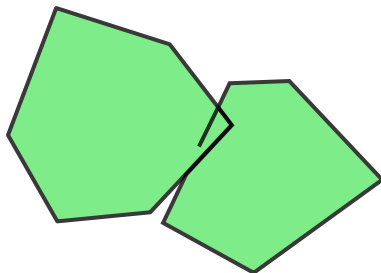
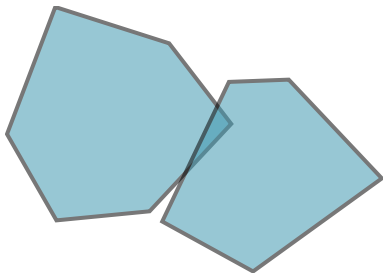
```
ERROR: Union returned a NULL geometry.
```

- problém jen u jednoho polygonu
- řešení samostatně pomocí `ST_SnapToGrid`

- použití funkce ST\_Difference
- odečtení překrytu od původního polygonu

```
SELECT A.gid,  
       ST_Multi(ST_Difference(  
                A.way, B.intersect  
                ))  
FROM f10.lesni_porosty AS A  
JOIN f10.lesy_intersect AS B  
ON A.gid = B.gid;
```

- ST\_Difference
- po odečtení zbyly nepatrné části překrytů
- některé polygony se tím znevalidnily



- zvětšení překrytu funkcí ST\_Buffer

```
SELECT A.gid,  
       ST_Multi(ST_Difference(  
               A.way, ST_Buffer(B.intersect, 0.01)  
               ))  
FROM f10.lesni_porosty AS A  
JOIN f10.lesy_intersect AS B  
ON A.gid = B.gid;
```

- kontrola validity nových geometrií
- vytvoření nové vrstvy lesů
  - UPDATE geometrií polygonů výsledkem rozdílů – 1293
  - DELETE polygonů, kde byl výsledek rozdílů  
GEOMETRYCOLLECTION EMPTY – 43
- kontrola nové vrstvy lesů
  - porovnání plochy starých a nových lesů
  - nové hledání překrytů a opakování postupu (3×)
  - počet překrytů se snižoval (15, 4, 3)
  - při posledním pokusu všechny překryty s nulovou plochou (již bez update)



- 1 Úvod
- 2 Validace lesů
  - Validace polygonů
  - Překryt polygonů
- 3 Ukázky dotazů
- 4 Shrnutí

## Otázka

Který újezd má největší rozlohu a kolik?

## Otázka

Který újezd má největší rozlohu a kolik?

```
SELECT name, ROUND(ST_Area(way)/1e6) AS area  
FROM f10.ujezdy AS ujezdy  
ORDER BY area DESC  
LIMIT 1;
```

## Otázka

Který újezd má největší rozlohu a kolik?

```
SELECT name, ROUND(ST_Area(way)/1e6) AS area
FROM f10.ujezdy AS ujezdy
ORDER BY area DESC
LIMIT 1;
```

Odpověď:

name	area
Vojenský újezd Hradiště	809

(1 row)

## Otázka

Který újezd má největší rozlohu a kolik?

```
SELECT name, ROUND(ST_Area(way)/1e6) AS area
FROM f10.ujezdy AS ujezdy
ORDER BY area DESC
LIMIT 1;
```

Odpověď:

name	area
Vojenský újezd Hradiště	809

(1 row)

Skutečná rozloha je 332  $km^2$ .

## Otázka

Kolik procent vojenského újezdu Hradiště pokrývá les?

## Otázka

Kolik procent vojenského újezdu Hradiště pokrývá les?

```
SELECT ROUND(((lesy_area/area)*100)::NUMERIC, 2)
FROM (
  SELECT SUM(ST_Area(ST_Intersection(ujezdy.way, lesy.way))) AS lesy_area,
         ST_Area(ujezdy.way) AS area
  FROM f10.ujezdy AS ujezdy
  JOIN f10.lesni_porosty AS lesy
       ON ujezdy.name = 'Vojensky ujezd Hradiste'
       AND ST_Intersects(ujezdy.way, lesy.way)
  GROUP BY ujezdy.way
) AS lesy_hrad;
```

## Otázka

Kolik procent vojenského újezdu Hradiště pokrývá les?

```
SELECT ROUND(((lesy_area/area)*100)::NUMERIC, 2)
FROM (
  SELECT SUM(ST_Area(ST_Intersection(ujezdy.way, lesy.way))) AS lesy_area,
         ST_Area(ujezdy.way) AS area
  FROM f10.ujezdy AS ujezdy
  JOIN f10.lesni_porosty AS lesy
       ON ujezdy.name = 'Vojensky ujezd Hradiste'
       AND ST_Intersects(ujezdy.way, lesy.way)
  GROUP BY ujezdy.way
) AS lesy_hrad;
```

Odpověď:

```
round
-----
16.86
```



- 1 Úvod
- 2 Validace lesů
  - Validace polygonů
  - Překryt polygonů
- 3 Ukázky dotazů
- 4 Shrnutí

- neočekávané problémy s funkcemi
  - ST\_Union
  - ST\_Difference
  - ST\_BuildArea
- nedostatečná dokumentace *PostGISu*, především hlášení chyb
  - nutnost hledat rady v diskusích – časově náročné
- výpočetní náročnost prostorových dotazů

## Výsledek

- validace lesů
- odstranění překrytů
- nad vrstvou je možné klást dotazy