

# PDT - PostGIS

[https://github.com/MennoCoehoorn/PDT\\_3.git](https://github.com/MennoCoehoorn/PDT_3.git)

Marek Štrba

November 2021

## Contents

- 1 Stiahnite a importujte si dataset pre Open Street mapy do novej DB 3
- 2 Zistite aké kraje sú na Slovensku (planet\_osm\_polygon, admin\_level = '4') a vypíšte ich súradnice ako text s longitude a latitude. 3
- 3 Zoradte kraje podľa ich veľkosti (st\_area). Veľkosť vypočítajte pomocou vhodnej funkcie a zobrazte v km<sup>2</sup> v SRID 4326. 3
- 4 Pridajte si dom, kde bývate ako polygón (nájdite si súradnice napr. cez google maps) do planet\_osm\_polygon (znova pozor na súradnicový systém). Výsledok zobrazte na mape 4
- 5 Zistite v akom kraji je váš dom 5
- 6 Pridajte si do planet\_osm\_point vašu aktuálnu polohu (pozor na súradnicový systém). Výsledok zobrazte na mape 5
- 7 Zistite či ste doma - či je vaša poloha v rámci vášho bývania 6
- 8 Zistite ako ďaleko sa nachádzate od FIIT (name = 'Fakulta informatiky a informačných technológií STU'). Pozor na správny súradnicový systém – vzdialenosť musí byť skutočná 7
- 9 Stiahnite si QGIS a vyplotujte kraje a váš dom z úlohy 2 na mape - napr. červenou čiarou 7
- 10 Zistite súradnice centroidu (čažiska) plošne najmenšieho okresu (vo výsledku nezabudnite uviesť aj EPSG kód súradnicového systému) 8
- 11 Vytvorte priestorovú tabuľku všetkých úsekov ciest, ktorých vzdialenosť od vzájomnej hranice okresov Malacky a Pezinok je menšia ako 10 km. 8
- 12 Jedným dopytom zistite číslo a názov katastrálneho územia (z dát ZBGIS), v ktorom sa nachádza najdlhší úsek cesty (z dát OSM) v okrese, v ktorom bývate 10
- 13 Vytvorte oblasť Okolie\_Bratislavы, ktorá bude zahŕňať zónu do 20 km od Bratislavы, ale nebude zahŕňať oblasť Bratislavы (Bratislava I až Bratislava V) a bude len na území Slovenska. Zistite jej výmeru 11

## 1 Stiahnite a importujte si dataset pre Open Street mapy do novej DB

Dáta som zo súboru načítal pomocou knižnice osm2pgsql.

```
root@DESKTOP-906700C:~# ./osm2pgsql -c ./FIIT/IM/1_Semester/OSM/slovakia.osm.pbf --database PDT_3 -U postgres --password -H localhost -P 5433 -r pbf
```

Figure 1

## 2 Zistite aké kraje sú na Slovensku (planet\_osm\_polygon, admin\_level = '4') a vypíšte ich súradnice ako text s longitude a latitude.

Polygon kraja som zredukoval na bod tým, že som vyrátal centroid pre každý kraj. Následne som pomocou funkcií ST\_X a ST\_Y zistil longitude a latitude. Hodnoty som transformoval do sústavy 4326 aby som si ich vedel overiť.

The screenshot shows a PostgreSQL Query Editor window. The query in the editor is:

```
1 SELECT name, ST_X(ST_Centroid(ST_Transform(way,4326))) AS longitude,
2        ST_Y(ST_Centroid(ST_Transform(way,4326))) AS latitude
3   FROM planet_osm_polygon WHERE admin_level = '4';
```

The results table below the query shows the following data:

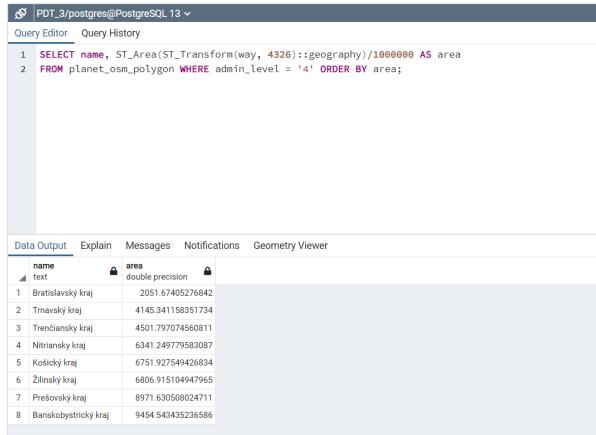
	name	longitude	latitude
1	Žilinský kraj	19.17732192085711	49.177529057683145
2	Košický kraj	21.26625694024157	48.697384329067326
3	Predovský kraj	21.224595461700968	49.123651728435625
4	Trenčiansky kraj	18.21338478197693	48.8584957424264
5	Banskobystrický kraj	19.50592492773542	48.51572816850621
6	Trnavský kraj	17.534436530917218	48.35301464227704
7	Bratislavský kraj	17.17906589756928	48.31741208636065
8	Nitriansky kraj	18.3108386472195	48.14201222531701

Figure 2

## 3 Zoradte kraje podľa ich veľkosti (st\_area). Veľkosť vypočítajte pomocou vhodnej funkcie a zobrazte v km<sup>2</sup> v SRID 4326.

Aby som hodnotu povrchu kraja dostať v km<sup>2</sup> tak som si prekonvertoval stĺpec way do sústavy 4326 a castol som ho ako typ geography. Takto mi St\_Area vyrátal výsledok v m<sup>2</sup>. Ten som predelil 1 000 000 a tak som dostał výsledok v

$\text{km}^2$ . Kraje som zoradil jednoducho cez ORDER BY s výsledkom hore-uvedeného výpočtu.



The screenshot shows a PostgreSQL query editor window titled "PDT\_3/postgres@PostgreSQL 13". The query editor tab is active, displaying the following SQL code:

```
1 SELECT name, ST_Area(ST_Transform(way, 4326)::geography)/1000000 AS area
2 FROM planet_osm_polygon WHERE admin_level = '4' ORDER BY area;
```

Below the code, the "Data Output" tab is selected, showing the results of the query:

name	area
Bratislavský kraj	2051.67405276842
Trnavský kraj	4145.34115851734
Trenčiansky kraj	4501.797074560811
Nitriansky kraj	6341.249779583087
Košický kraj	6751.927549426834
Žilinský kraj	6806.915104947965
Presovský kraj	8971.630508024711
Banskobystrický kraj	9454.543435236586

Figure 3

#### 4 Pridajte si dom, kde bývate ako polygón (nájdite si súradnice napr. cez google maps) do planet\_osm\_polygon (znova pozor na súradnicový systém). Výsledok zobrazte na mape

Súradnice rohov mojej bytovky som si zistil z google maps. Keďže sú ale v 4326 tak som si cez ST\_Transform vytvorený polygón zmenil do sústavy 3857. Výsledok som potom trasformoval cez ST\_AsText premenil na text aby som si cez polygon vytvoril pri inserte z neho záznam. Tento postup som zvolil, lebo som nevedel prísť na to ako inak vytvoriť zápis do databázy tak aby som mu mohol určiť aj iný stĺpec, nie len way. Výsledný polygon som potom zobrazil na mape ako geography.

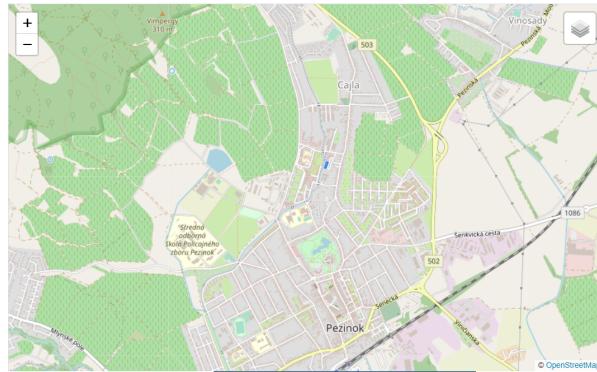


Figure 4

## 5 Zistite v akom kraji je váš dom

Cez admin\_level stĺpec som si odfiltroval iba kraje a zároveň som vybral kraj, ktorý obsahoval môj dom cez príkaz ST\_Contains.

```
PDT_3/postgres@PostgreSQL 13
Query Editor Query History
1 SELECT name FROM planet_osm_polygon WHERE admin_level = '4'
2 AND ST_Contains(way, (SELECT way FROM planet_osm_polygon WHERE name = 'Domov'));
```

name	text
Bratislavský kraj	

Figure 5

## 6 Pridajte si do planet\_osm\_point vašu aktuálnu polohu (pozor na súradnicový systém). Výsledok zobrazte na mape

Podobne ako pri pridávaní bývania som si zistil súradnice pomocou Google Maps a pred insertom som ich pretransformoval zo sústavy 3857 na 4326.

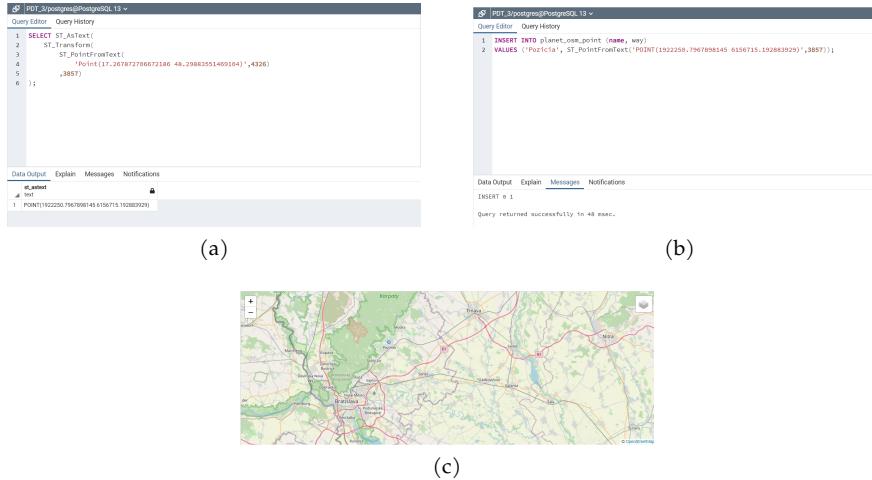


Figure 6: (a) - Konverzia z 4326 na 3857 , (b) - Insert do DB, (c) - výsledok zobrazený na mape

## 7 Zistite či ste doma - či je vaša poloha v rámci vášho bývania

Rovnako ako pri zisťovaní v ktorom kraji sa nachádza môj dom som použil funkciu ST\_Contains. Svoj dom som získal cez hlavný select a aktuálnu pozíciu cez select ako subquery.



Figure 7

## 8 Zistite ako ďaleko sa nachádzate od FIIT (name = 'Fakulta informatiky a informačných technológií STU'). Pozor na správny súradnicový systém – vzdialenosť musí byť skutočná

Vzdialenosť som vyrátal cez funkciu ST\_Distance. Podobne ako v predošej úlohe som jeden z geometrickych údajov získal v hlavnej query a druhý v select subquery. Aby som zabezpečil presnosť merania tak som oba údaje transformoval do sústavy 4326 a výsledok ST\_Distance som castol ako geography. Výsledok som chcel dostať v km tak som ho na záver predelil 1000.

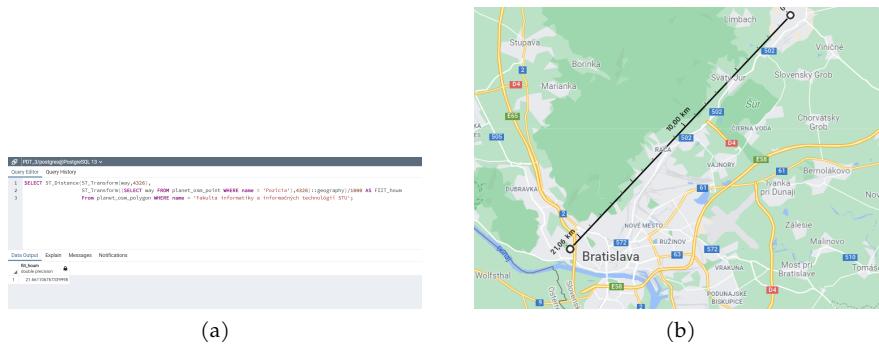


Figure 8: (a) - Vyrátaná vzdialenosť , (b) - Približná kontrola na google maps

## 9 Stiahnite si QGIS a vyplotujte kraje a váš dom z úlohy 2 na mape - napr. červenou čiarou

Select pre túto úlohu bol jednoduchý. Stačilo vybrať všetky polygóny, ktoré majú admin\_level = '4' alebo meno 'Domov'

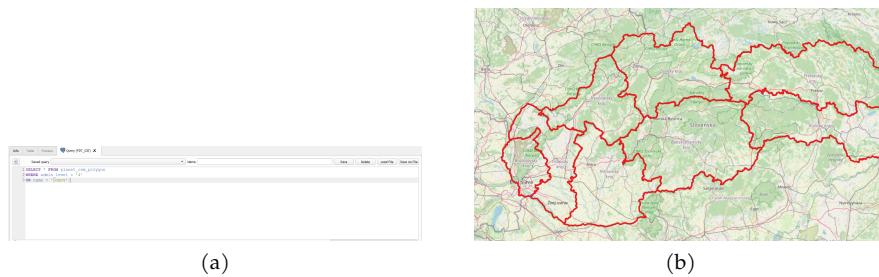


Figure 9: (a) - Select , (b) - Zobrazenie na mape

## 10 Zistite súradnice centroidu (čažiska) plošne najmenšieho okresu (vo výsledku nezabudnite uviesť aj EPSG kód súradnicového systému)

Ako prvé som si vytvoril query, ktorá mi vráti všetky Slovenské okresy. Výsledok som si overil v qgise. Touto query som zároveň zistil, že dataset neobsahuje okresy Košice II a IV, to ale nevadilo, keďže najmenší okres na Slovensku je okres Bratislava I. Plochu okresu som vypočítal z dát transformovaných do sústavy 4326 a castnutých ako geography aby bola v  $m^2$ . Plochu som predelili 1 000 000 aby bola v  $km^2$ . Podľa tejto plochy som následne pomocou ORDER BY ASC LIMIT 1 vybral okres s najmenšou plochou. Do selecetu som doplnil výpočet longitude a latitude čažiska okresu cez ST\_Y, ST\_X a ST\_Centroid. Súradnice čažiska sú v systéme EPSG 4326.

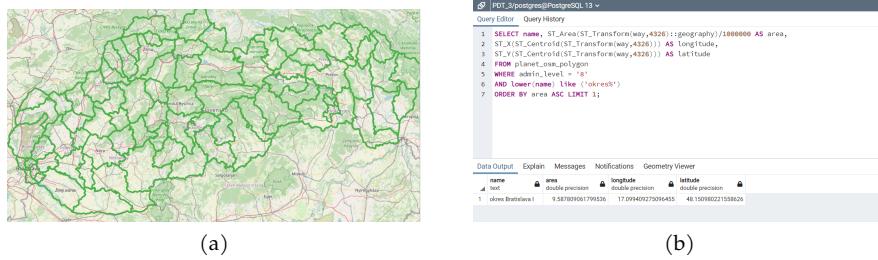


Figure 10: (a) - Overenie v QGise , (b) - Výsledná query

## 11 Vytvorte priestorovú tabuľku všetkých úsekov ciest, ktorých vzdialenosť od vzájomnej hranice okresov Malacky a Pezinok je menšia ako 10 km.

Ako prvé som si vytvoril 10km buffer okolo hraníc okresu Pezinok s okresom Malacky a tento buffer som si pre ďalšie použitie uložil do DB ako MPK\_Buffer polygon.

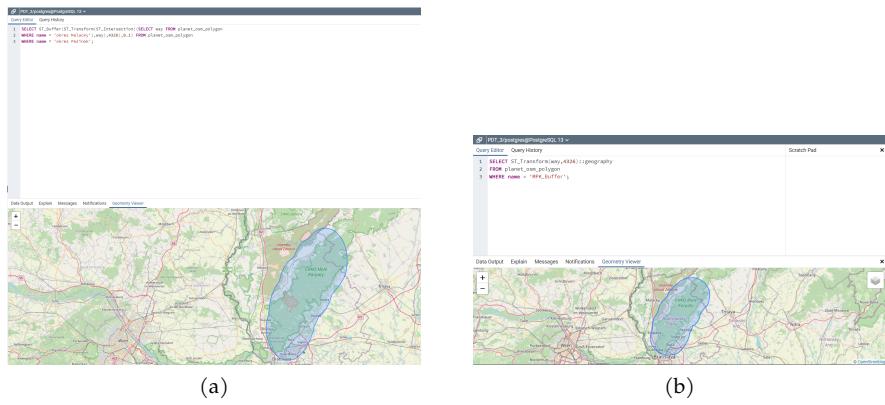


Figure 11: (a) - Tvorba bufferu , (b) - Overenie správnosti bufferu v DB

Následne som z planet.osm.roads vybral všetky prieseky ciest s vytvoreným bufferom pomocou funkcie ST\_Intersection. To že išlo o cestu som overil tým, že som kontroloval, či roads.highway nie je Null, keďže podľa OSM dokumentácie sa tento stĺpec používa na označenie všetkých ciest. Vytvoril som si tabuľku mpk.roads, ktorej som pridal geometricky column way. Do tejto tabuľky som ako linestring uložil výsledky intersectionov. Pre overenie som na mape zo-brazil výsledok selectu nad celou tabuľkou.

The figure consists of five panels labeled (a) through (e), each showing a PostgreSQL/PostGIS query editor window.

- (a)**: Shows a map of Austria with a blue buffer polygon around a specific area. The SQL query is:

```
1 SELECT ST_Transform(ST_Intersection(poly.wkb, roads.wkb),4326)::geography
2 FROM planet_osm_polygon poly,planet_osm_roads roads
3 WHERE poly.name = 'M99_Buffer' AND roads.highway IS NOT NULL;
```

- (b)**: Shows the creation of a new table named apk\_roads. The SQL query is:

```
1 CREATE TABLE apk_roads(
2   id SERIAL
3 );
```

- (c)**: Shows the addition of a geometry column to the apk\_roads table. The SQL query is:

```
1 SELECT AddGeometryColumn('apk_roads', 'way', 3857, 'MultiLineString', 2);
```

- (d)**: Shows the insertion of data into the apk\_roads table. The SQL query is:

```
1 INSERT INTO apk_roads (way)
2 SELECT ST_Multi(ST_Intersection(poly.wkb, roads.wkb))
3 FROM planet_osm_polygon poly,planet_osm_roads roads
4 WHERE poly.name = 'M99_Buffer' AND roads.highway IS NOT NULL;
```

The message "Query returned successfully in 8 secs 941 msec." is displayed at the bottom.

- (e)**: Shows a map of Austria with the newly inserted道路 (roads) layer visible as a blue line.

Figure 12: (a) - Select všetkých úsekov ciest v danom buffery , (b) - Vytvorenie tabuľky, (c) - Pridanie vhodného geometrického stĺpca, (d) - Zápis ciest do tabuľky, (e) - Kontrola na mape

## 12 Jedným dopytom zistite číslo a názov katastrálneho územia (z dát ZBGIS), v ktorom sa nachádza najdlhší úsek cesty (z dát OSM) v okrese, v ktorom bývate

Dáta som si do databázy načítal pomocou gdal knižnice a príkazu ogr2ogr:



Figure 13

Pri dopyte na zistenie katastrálnych území som postupol tak, že som s najprv vytvoril dopyt na najdlší úsek cesty v mojom okrese (okres Pezinok). Najdlší úsek som dostal po vyrádani ST\_Length ST\_Interersectionu medzi roads.way a polygon.way, pričom polygon bol okres Pezinok. To že išlo o cestu som overil tým, že som kontroloval, či roads.highway nie je Null, keďže podľa OSM dokumentácie sa tento stĺpec používa na označenie všetkých ciest. Tieto výsledky dopytu som zoradil podľa tejto dĺžky, limitoval na 1 a tak som dostal najdlší úsek cesty v mojom okrese. Z tohto poddopytu som si vrátil aj samotný intersection medzi cestou a okresom, ktorý som následne pomocou CTE (with) prenesol do hlavnej query. V hlavnej query som si potom doselectoval údaje z tabuľky katastrálnych území a cez ST\_Intersects som zistil, s ktorým územím má môj úsek cesty prienik.

```
PDT_3@PostgreSQL:~
```

Query Editor    Query History

```
1 WITH longest_road AS (SELECT roads.name, ST_Intersection(poly.way), roads.way) AS road,
2      ST_Length(STIntersection(poly.way, roads.way)) AS len
3   FROM planet_osm_polygon poly, planet_osm_roads roads
4  WHERE poly.name = 'ökers Pezinok' AND roads.highway IS NOT NULL
5 ORDER BY len DESC
6 LIMIT 1)
7 SELECT kata.idn5, kata.nms5
8 FROM ku_0 kata, longest_road l
9 WHERE ST_Intersects(STTransform(kata.shape,3857), l.road);
```

Figure 14

13 Vytvorte oblasť Okolie\_Bratislavu, ktorá bude zahŕňať zónu do 20 km od Bratislavu, ale nebude zahŕňať oblasť Bratislavu (Bratislava I až Bratislava V) a bude len na území Slovenska. Zistite jej výmeru

Samotný buffer som vytvoril po krokoch. Ako prvé som vytvoril 20km buffer okolo Bratislavы (admin\_level = '6'). Následne som tento buffer limitoval cez intersection na Slovensko. Ako posledné som vyňal Bratislavu tak, že som pomocou ST\_MakePolygon a ST\_ExteriorRing z bufferu, ktorý som predtým vytvoril danú oblasť Bratislavы dal ako vnútorný výrez.

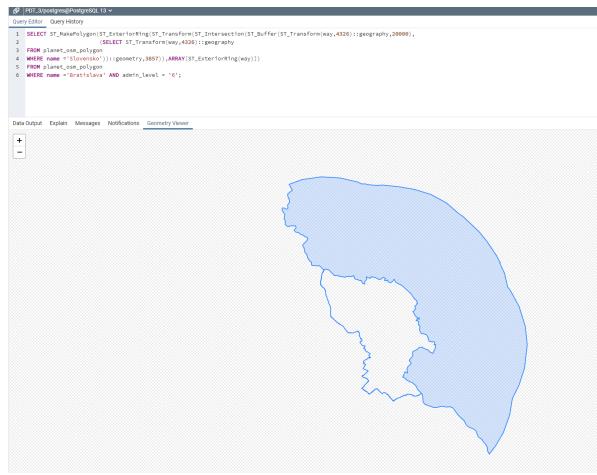


Figure 15

Výsledok som si uložil do tabuľky planet\_osm\_polygon ako 'Okolie\_Bratislav'. Výmeru som potom jednoducho vyrátal pomocou ST\_Area a ST\_Transform. Výsledok je 1480,1 km<sup>2</sup>.

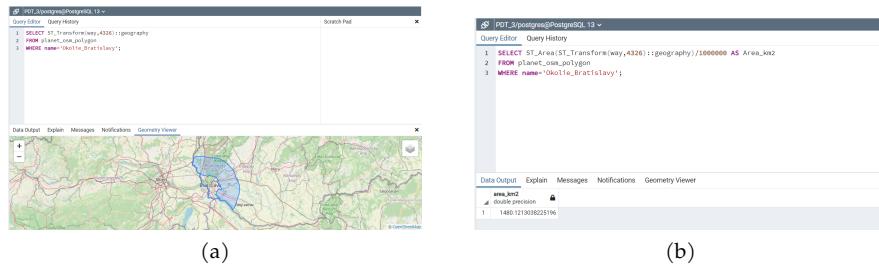


Figure 16: (a) - Zobrazenie na mape , (b) - Výsledok