

Fakulta informatiky a informačných technológií STU v Bratislave

Zadanie 3 – PostGIS

29.9.2022

Matej Delinčák

Github classroom: <https://github.com/FIIT-DBS/zadanie-pdt-mateju25>

1. stiahnite a importujte si dataset pre Open Street mapy z <https://download.geofabrik.de/europe/slovakia.html> do novej DB

Vytvoril som extension na postgis.

```
1 CREATE EXTENSION postgis;
2 CREATE EXTENSION postgis_topology;
```

A pomocou *osm2pgsql* som importoval dáta z open-street map do databázy:

```
osm2pgsql.exe -c -r pbf -d maps -U postgres -W -H localhost -P 5432 -v slovakia-latest.osm.pbf
```

Dĺžka trvania:

```
[0] osm2pgsql took 151s (2m 31s) overall.
```

Veľkosti tabuliek:

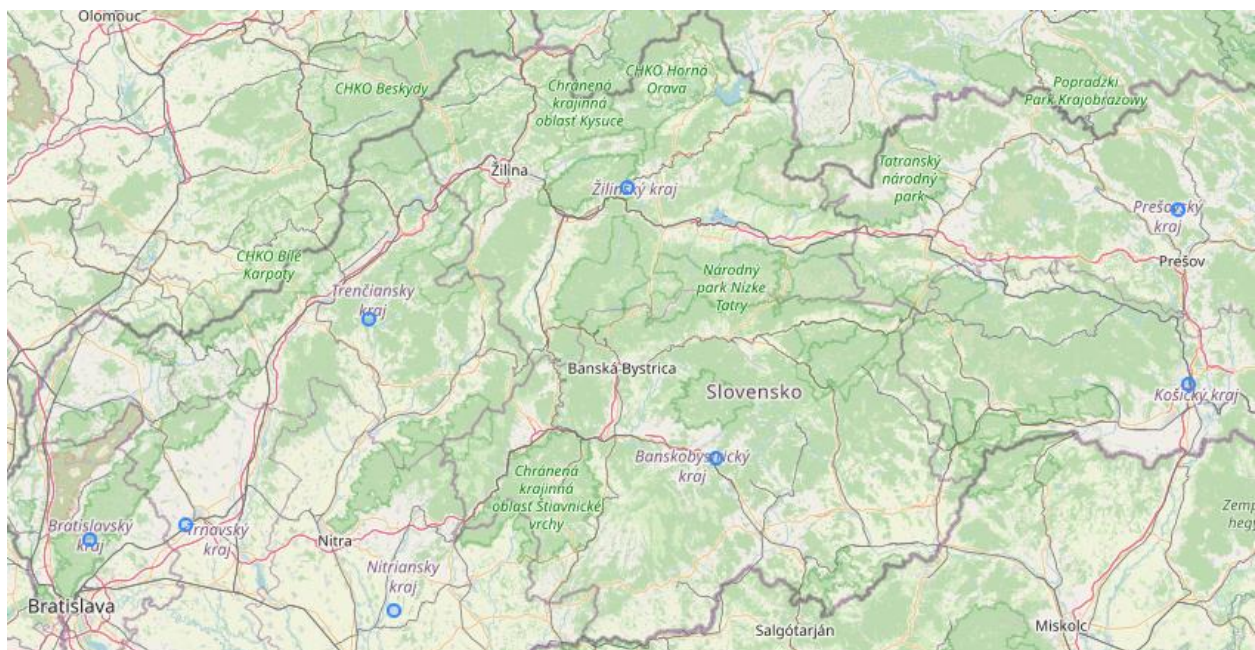
Table name	Tuples inserted
planet_osm_line	890,884
planet_osm_point	767,598
planet_osm_polygon	2,655,609
planet_osm_roads	70,640
spatial_ref_sys	8,500

2. zistíte aké kraje sú na Slovensku (planet_osm_polygon, admin_level = '4') a vypíšete ich súradnice ako text s longitude a latitude.

Pomocou centroidu som vyrátal pomyselný stred každého kraja a jeho súradnice som pretypoval na správny súradnicový systém. Je to preto, aby som si mohol zobrazíť mapku a potom pomocou ST_X a ST_Y vytiahol zemepisnú dĺžku a výšku.

```
SELECT name as meno,  
ST_X(ST_Transform(ST_Centroid(way), 4326)) AS zemepisna_dlzka,  
ST_Y(ST_Transform(ST_Centroid(way), 4326)) AS zemepisna_sirka  
FROM planet_osm_polygon  
WHERE admin_level = '4';
```



	meno text	zemepisna_dlzka double precision	zemepisna_sirka double precision
1	Žilinský kraj	19.177320070747566	49.17752681312351
2	Prešovský kraj	21.224596605750712	49.12365238366032
3	Banskobystrický kraj	19.503924241301327	48.515727802946856
4	Košický kraj	21.266252544468706	48.69738433926061
5	Bratislavský kraj	17.17906401203241	48.31741628252674
6	Trenčiansky kraj	18.213384566226075	48.85849547508841
7	Trnavský kraj	17.53483556069515	48.35301540208069
8	Nitriansky kraj	18.310838342824315	48.14201199419118



3. zoradíte kraje podľa ich veľkosti (st_area). Veľkosť vypočítajte pomocou vhodnej funkcie a zobrazte v km² v SRID 4326.

Pomocou ST_AREA som vyrátal veľkosť, ktorá bola v metroch štvorcových a teda bolo ju teda ešte prenásobiť. Po ST_TRANSFORM bolo treba dáta dostať z geometry typu na geography aby výsledok bol v metroch.

```
SELECT name as meno,  
CONCAT(ST_AREA(ST_Transform(way, 4326)::geography)/1000000, ' km^2') AS velkost  
FROM planet_osm_polygon  
WHERE admin_level = '4'  
ORDER BY velkost
```

	 meno text	 velkost text
1	Bratislavský kraj	2051.660249334432 km^2
2	Trnavský kraj	4145.350058274841 km^2
3	Trenčiansky kraj	4501.795536203455 km^2
4	Nitriansky kraj	6341.240667758643 km^2
5	Košický kraj	6751.964233845743 km^2
6	Žilinský kraj	6806.886696528445 km^2
7	Prešovský kraj	8971.62395364223 km^2
8	Banskobystrický kraj	9454.546380914244 km^2

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.

Cez google maps som našiel súradnice domu. Pomocou ST_GeomFromGeoJSON som z nich vytvoril polygón a zmenil som na správny súradnicový systém v tabuľke.

```
INSERT INTO planet_osm_polygon (osm_id, name, z_order, way_area, way)
VALUES (1106441030, 'my_house', 0, 0, ST_TRANSFORM(ST_GeomFromGeoJSON('{
  "type": "Polygon",
  "coordinates": [[
    [18.785275, 49.441545],
    [18.785360, 49.44144936964138],
    [18.785019, 49.441331647065105],
    [18.784938, 49.4414301852733],
    [18.785275, 49.441545],
  ]],
  "crs": {"type": "name", "properties": {"name": "EPSG:4326"}}
}')), 3857));
```



5. zistite v akom kraji je váš dom.

Vyfiltroval som si polygóny aby som mal len kraje a potom pomocou ST_Contains som porovnal, či sa môj polygón nachádza v rámci polygónov krajov.

```
SELECT name
FROM planet_osm_polygon
WHERE admin_level = '4' AND ST_Contains(way, (SELECT way FROM planet_osm_polygon WHERE name = 'my_house'));
```

	name text	
1	Žilinský kraj	

**6. pridajte si do planet_osm_point vašu aktuálnu polohu (pozor na súradnicový systém).
Výsledok zobrazte na mape.**

Opäť pomocou google maps som si našiel polohu a cez ST_GeomFromText som vytvoril bod, ktorému som potom zmenil súradnicový systém na taký, ktorý sa nachádza v tabuľke.

```
INSERT INTO planet_osm_point (osm_id, name, way)
VALUES (10124872640, 'my_location', ST_TRANSFORM(ST_GeomFromText('POINT(17.06431 48.15805)', 4326),3857));
```



7. zistite či ste doma - či je vaša poloha v rámci vášho bývania.

Rovnako ako v úlohe 5 som použil ST_Contains. Z tabuľky pointov som si subselectom vybral svoju pozíciu a z polygónov som si vyfiltroval svoju bytovku.

```
SELECT ST_Contains(way, (SELECT way FROM planet_osm_point WHERE name = 'my_location'))
FROM planet_osm_polygon
WHERE name = 'my_house'
```

	st_contains
	boolean
1	false

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

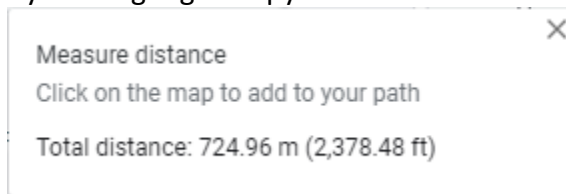
Pomocou ST_DISTANCE som vyratal vzdialenost stredy fakulty a mojej pozicie. Vysledok som previedol na SRID: 26986 Massachusetts state plane meters, čo podľa dokumentácie dáva presné výsledky a výsledok je v metroch. Hodnota sa skoro presne zhoduje s porovnaním na google mapách.

```
SELECT ST_DISTANCE(  
    ST_TRANSFORM(ST_CENTROID(way), 26986)::geometry,  
    (SELECT ST_TRANSFORM(way, 26986) FROM planet_osm_point WHERE name = 'my_location')::geometry  
)  
FROM planet_osm_polygon  
WHERE name = 'Fakulta informatiky a informačných technológií STU'
```

Výsledok query:

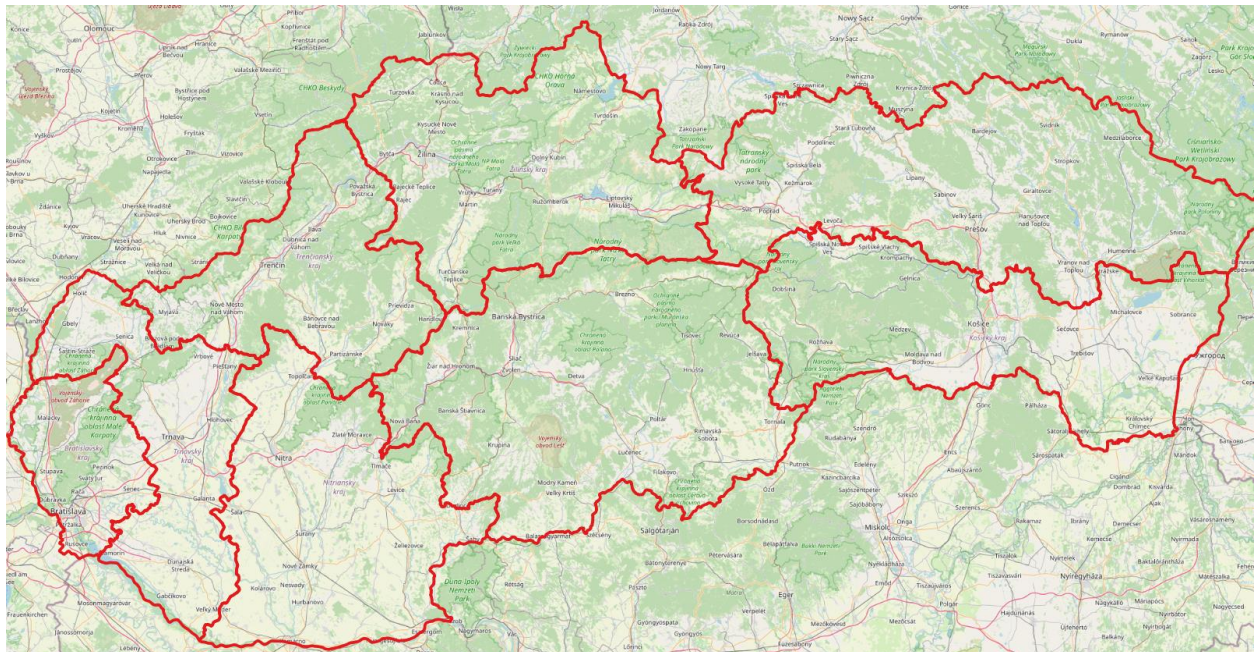
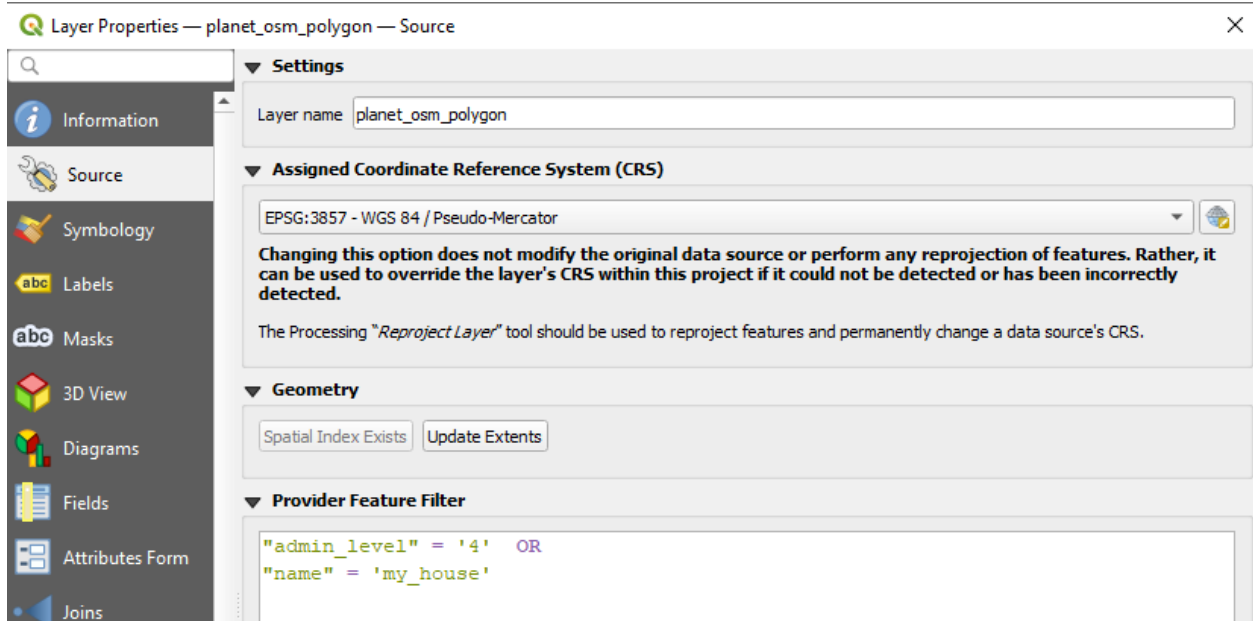
st_distance double precision	
1	727.0448412721971

Výsledok google mapy:



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

Vyrobil som si layer z tabuľky planet_osm_polygon a dorobil som si feature filter.



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

V tejto query som využil funkcie z predchádzajúcich úloh na nájdenie centroidu a plochu polygónu ako aj jeho súradnice. Stĺpec SRID je srid údajov v tabuľke, nie zobrazených dát. SRID zobrazených dát je 4326. Pre admin_level som použil hodnotu 8, lenže mi to našlo aj polygóny mimo Slovenska. Tieto som ručne našiel a odstránil z výsledku.

```
SELECT name as meno,  
ST_SRID(way) as srid,  
ST_AREA(ST_Transform(way, 4326)::geography)/1000000 AS velkost,  
ST_X(ST_Centroid(ST_Transform(way, 4326))) AS zemepisna_dlzka,  
ST_Y(ST_Centroid(ST_Transform(way, 4326))) AS zemepisna_sirka  
FROM planet_osm_polygon  
WHERE admin_level = '8' and name not in ('Vének', 'Szentháromság-templom', 'Lába  
ORDER BY velkost LIMIT 1;
```

	meno text	srid integer	velkost double precision	zemepisna_dlzka double precision	zemepisna_sirka double precision
1	okres Bratislava I	3857	9.588613370524051	17.099410126661734	48.150979076236325



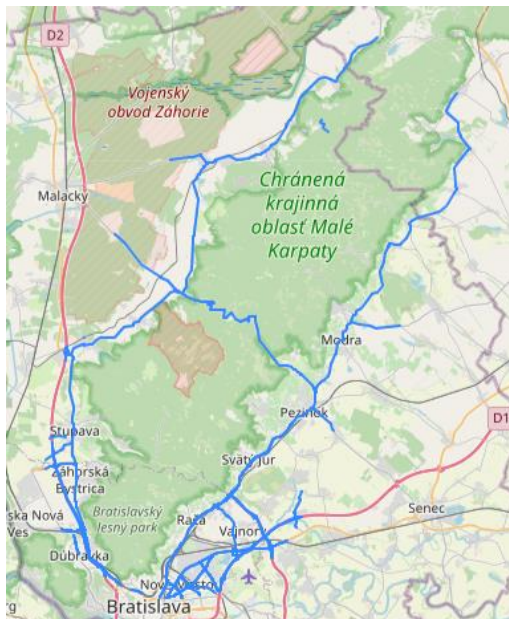
11. Vytvorte priestorovú tabuľku všetkých úsekov ciest, ktorých vzdialenosť od vzájomnej hranice okresov Malacký a Pezinok je menšia ako 10 km.

Najskôr som si vytvoril polygon cez ST_Buffer, kde som pomocou intersection vytvoril prienik hraníc a ten som transformoval na systém 4326. Premenu na 4326 som urobil preto, lebo som našiel, že 3857 je nepresný pre rátanie vzdialeností.

```
INSERT INTO planet_osm_polygon (name, way) VALUES ('buffer', (
SELECT ST_TRANSFORM (ST_BUFFER (ST_TRANSFORM (ST_INTERSECTION (way,
(SELECT way FROM planet_osm_polygon WHERE admin_level = 8::text AND name like '%Malacky%')), 4326)::geography, 10000)::geometry, 3857)
FROM planet_osm_polygon WHERE admin_level = 8::text AND name like '%Pezinok%'))
```

Následne som si z tabuľky planet_osm_roads vytiahol všetky záznamy a pomocou intersection s mojim bufferom som vytiahol všetky potrebné cesty. Musel som ale vyfiltrovať MultiLineString, aby mi tam ostali len LineString typy. Trebalo ešte vyfiltrovať iba cesty, lebo v roads sa nachádzali aj iné objekty. Podľa dokumentácie je cesta to, kde bolo v highway hocičo okrem NULL.

```
SELECT way FROM (SELECT ST_INTERSECTION (way, (SELECT way FROM planet_osm_polygon WHERE name = 'buffer')) as way,
ST_GeometryType(ST_INTERSECTION (way, (SELECT way FROM planet_osm_polygon WHERE name = 'buffer')))) as name_new
FROM planet_osm_roads WHERE highway LIKE '%' t
WHERE name_new = 'ST_LineString'
```



A tieto záznamy som vložil do tabuľky buffer_roads a záznamy som vložil do nej.

```
CREATE TABLE buffer_roads (
  id bigint not null constraint buffer_roads_pkey primary key,
  way geometry(MultiLineString, 3857)
)
```

```
INSERT 0 30678
```

Query returned successfully in 1 secs 982 msec.

12. Jedným dopytom zistíte číslo a názov katastrálneho územia (z dát ZBGIS, https://www.geoportal.sk/sk/zbgis_smd/na-stiahnutie/), v ktorom sa nachádza najdlhší úsek cesty (z dát OSM) v okrese, v ktorom bývate.

```
ogr2ogr -f PostgreSQL "PG:dbname=maps user=postgres password=_____" hranice.gpkg
```

Zo stránky som si stiahol dáta a pomocou ogr2ogr som dáta importoval do databázy. Prvá časť query mi vytvorí view, v ktorom sa nachádza vyselektovaná najdlhšia cesta v mojom okrese. A následne dám intersect na dáta z ZGBIS-u a vyselectujem id a názov katastrálnych území, v ktorom sa nachádza.

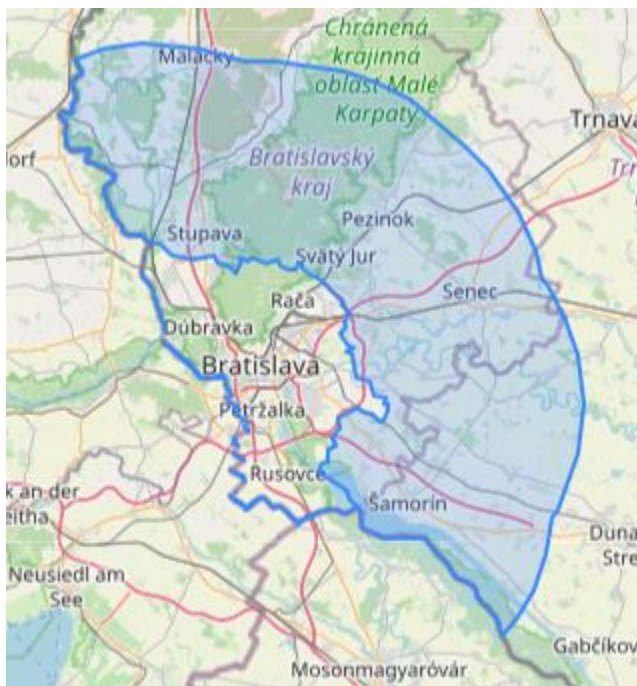
```
WITH road AS (  
  SELECT ST_INTERSECTION(p.way, r.way) as way, ST_LENGTH(ST_INTERSECTION(p.way, r.way)) as length  
  FROM planet_osm_polygon p, planet_osm_roads r WHERE p.admin_level = 8::text AND p.name like '%Čadca%' AND r.highway LIKE '%'  
  ORDER BY length DESC LIMIT 1)  
SELECT ku_0.objectid, ku_0.idn5, ku_0.nm5, ku_0.idn4, ku_0.nm4, ku_0.idn3, ku_0.nm3  
FROM ku_0, road  
WHERE ST_INTERSECTS(ST_TRANSFORM("Shape",3857), road.way)
```

	objectid [PK] integer	idn5 integer	nm5 character varying (50)	idn4 integer	nm4 character varying (80)	idn3 smallint	nm3 character varying (80)
1	5232	834858	Makov	509299	Makov	502	Čadca

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

Pomocou bufferu a intersect som si vyrobil polygón okolo Bratislavy vrátane nej. A následne som od toho pomocou ST_DIFFERENCE odrátal iba Bratislavu.

```
INSERT INTO planet_osm_polygon (name, way) VALUES ('Okolie_Bratislavy', (
WITH
buffer as (
SELECT ST_Intersection(ST_BUFFER (ST_TRANSFORM (way, 4326)::geography, 20000),
(SELECT ST_TRANSFORM (way, 4326) FROM planet_osm_polygon WHERE name = 'Slovensko' AND admin_level = '2')) as way
FROM planet_osm_polygon WHERE name = 'Bratislava' AND admin_level = '6'
),
bratislava as (
SELECT ST_TRANSFORM (way, 4326) as way FROM planet_osm_polygon WHERE name = 'Bratislava' AND admin_level = '6'
)
SELECT ST_Difference(bf.way::geometry, br.way::geometry) FROM buffer bf, bratislava br))
```



Výmeru som vyrátal rovnako ako v úlohach 3 a 10.

```
SELECT CONCAT(ST_AREA(ST_TRANSFORM (way, 4326)::geography)/1000000, ' km^2') as vymera
FROM planet_osm_polygon WHERE name = 'Okolie_Bratislavy'
```

	vymera text	
1	1480.1277491355397 km^2	