Zadanie 3 – PostGIS

Odovzdanie do 7.11.2021 23:59 – máte na to 2 týždne – dostanete za to 10 bodov.

Otázky 2-9 sú dokopy za 6 bodov (každá 0,75). Zadania 10 - 14 je za 4 body (každá 1).

Zadania prosím neopisujte jednoslovne ale zmysluplnou vetou (nie slohová práca - teda vecne). Vždy priložte screenshot z tabuľky alebo mapy, alebo oba. Mapu vám vie vykresliť aj pgadmin.

Pozn. - opisoval som iba odpovede kde mi to prišlo potrebné

github - https://github.com/jaruji/PDT-3

1. stiahnite a importujte si dataset pre Open Street mapy z http://download.freemap.sk/slovakia.osm/slovakia.osm.pbf do novej DB

```
2021-10-27 18:23:37  osm2pgsql version 1.5.1
Password:postgres
2021-10-27 18:23:38  Database version: 13.4
2021-10-27 18:23:38  PostGIS version: 3.1
2021-10-27 18:23:39  Setting up table 'planet_osm_point'
2021-10-27 18:23:39  Setting up table 'planet_osm_line'
2021-10-27 18:23:39  Setting up table 'planet_osm_polygon'
2021-10-27 18:23:39  Setting up table 'planet_osm_roads'

2021-10-27 18:33:12  Reading input files done in 573s (9m 33s).
2021-10-27 18:33:12    Processed 35349271 nodes in 254s (4m 14s) - 139k/s
2021-10-27 18:33:12    Processed 4555460 ways in 261s (4m 21s) - 17k/s
2021-10-27 18:33:12    Processed 57968 relations in 58s - 999/s
2021-10-27 18:33:13  Clustering table 'planet_osm_line' by geometry...
2021-10-27 18:33:14  Clustering table 'planet_osm_roads' by geometry...
2021-10-27 18:33:14  Clustering table 'planet_osm_point' by geometry...
2021-10-27 18:33:14  Clustering table 'planet_osm_polygon' by geometry...
2021-10-27 18:38:14  Creating geometry index on table 'planet_osm_roads'...
2021-10-27 18:38:20  Creating osm_id index on table 'planet_osm_roads'...
2021-10-27 18:38:26  Analyzing table 'planet_osm_roads'...
2021-10-27 18:38:27  Done postprocessing on table 'planet_osm_nodes' in 0s
2021-10-27 18:38:28  Building index on table 'planet_osm_ways'
2021-10-27 18:40:26  Creating geometry index on table 'planet_osm_point'...
2021-10-27 18:41:08  Creating osm_id index on table 'planet_osm_point'...
2021-10-27 18:41:25  Analyzing table 'planet_osm_point'...
2021-10-27 18:41:26  Building index on table 'planet_osm_rels'
2021-10-27 18:42:07  Creating geometry index on table 'planet_osm_line'...
2021-10-27 18:42:45  Creating osm_id index on table 'planet_osm_line'...
2021-10-27 18:43:36  Analyzing table 'planet_osm_line'...
2021-10-27 18:46:35  Done postprocessing on table 'planet_osm_ways' in 487s (8m
7s)
2021-10-27 18:46:35  Done postprocessing on table 'planet_osm_rels' in 10s
2021-10-27 18:46:35  All postprocessing on table 'planet_osm_point' done in 492s
 (8m 12s).
2021-10-27 18:46:35  All postprocessing on table 'planet_osm_line' done in 623s
(10m 23s).
2021-10-27 18:47:28  Creating geometry index on table 'planet_osm_polygon'...
2021-10-27 18:48:40  Creating osm_id index on table 'planet_osm_polygon'...
2021-10-27 18:48:52  Analyzing table 'planet_osm_polygon'...
2021-10-27 18:48:52  All postprocessing on table 'planet_osm_polygon' done in 93
8s (15m 38s).
2021-10-27 18:48:52  All postprocessing on table 'planet_osm_roads' done in 314s
 (5m 14s).
2021-10-27 18:48:52  osm2pgsql took 1514s (25m 14s) overall.
```

```
prikaz - osm2pgsql -U postgres -W -d gis -s -P 5433 --hstore
--number-processes 4 -C 20480 slovakia.osm.pbf
```

Použil som osm2pgsql a príkaz napísaný vyššie na import dát do databázy pg.

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

Nevedel som ci mam napisat suradnice centroidu alebo suradnice co su vo way iba konvertovane (ci mal byt vysledok list vsetkych long lat polygonu alebo iba jeden bod,

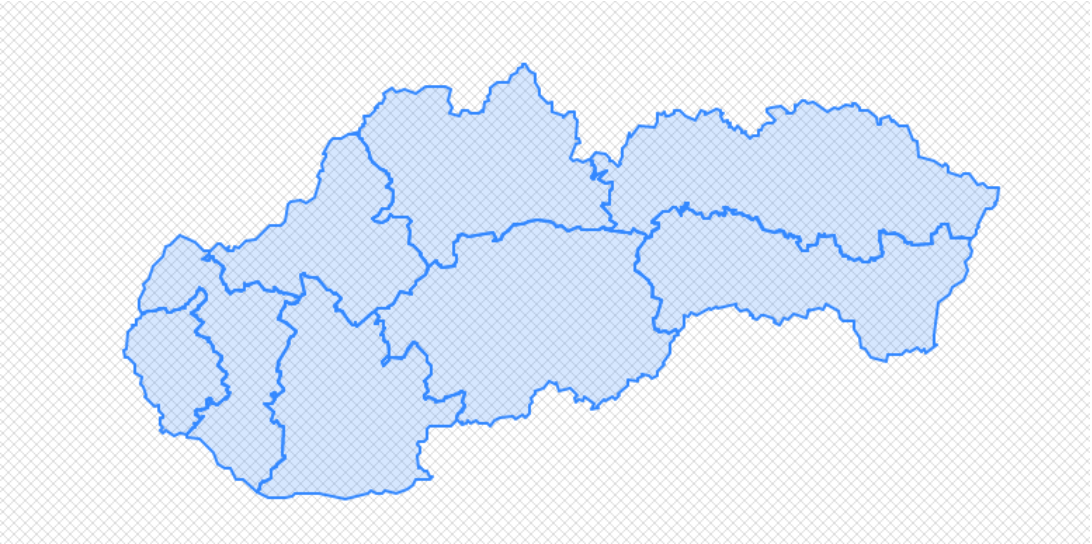
preto centroid), tak som spravil obe.

```
SELECT name, ST_ASTEXT(ST_TRANSFORM(ST_CENTROID(pog.way), 4326)) as long_lat
FROM planet_osm_polygon as pog WHERE admin_level='4';
```

	name text	long_lat text
1	Žilinský kraj	POINT(19.177181796454036 49.178557983144124)
2	Prešovský kraj	POINT(21.22418274002467 49.124325678884055)
3	Košický kraj	POINT(21.26574303407972 48.69795092669973)
4	Nitriansky kraj	POINT(18.31072331607922 48.143596531625775)
5	Banskobystrický kraj	POINT(19.50410244475429 48.517016563021855)
6	Trnavský kraj	POINT(17.53422218070288 48.355443952532966)
7	Trenčiansky kraj	POINT(18.213524759741162 48.859445030834856)
8	Bratislavský kraj	POINT(17.178989849285298 48.318001482710045)

```
SELECT name, ST_ASTEXT(ST_TRANSFORM(way, 4326)) FROM planet_osm_polygon WHERE admin_level = '4'
```

	name text	st_astext text
1	Košický kraj	POLYGON((20.181221999999998 48.74096699975657,20.181416 48.740548999975667,20.181695999999995 48.73991799975683,20.181484999999995 48.73968199975689,20.181665000000002 48.739227999757006,20.181813 48.7388609997571,20.182119000000
2	Banskobystrický kraj	POLYGON((18.478672999999997 48.54860199980419,18.478683 48.54857799980419,18.479153 48.548347999804236,18.479732999999996 48.548080999980432,18.480818999999997 48.54763199980441 6,18.481864 48.54716199980454,18.482448 48.5469249998
3	Trenčiansky kraj	POLYGON((17.353035699999996 48.77752659974 75,17.353104 48.77746609974752,17.3539395 48.77677139974771,17.354356199999998 48.77641779974778,17.355570299999997 48.775404499748035,17.3563613 48.77474639974818,17.3566281 48.774519699
4	Bratislavský kraj	POLYGON((16.833189099999995 48.380524099845566,16.8332193 48.380208499845644,16.833291399999997 48.3799262998457,16.8334366 48.37961889984577,16.8344514 48.37821819984612,16.8374269 48.37498819984691,16.8382026 48.37449799984704,1
5	Trnavský kraj	POLYGON((16.933595 48.600625999791326,16.933598999999997 48.60019099979144,16.933632999999997 48.59973599979154,16.933687 48.59934299979165,16.933809 48.59893499979174 4,16.933974 48.59851499979186,16.934186 48.59809599979196,16.934
6	Nitriansky kraj	POLYGON((17.705419 47.75898499999446,17.705507699999995 47.75889909999449,17.707337999999996 47.75735999999485,17.708578999999997 47.75667799999499,17.709914 47.75608199999513,17.711332 47.75557999999525,17.712815999999997 47.755
7	Žilinský kraj	POLYGON((18.322605 49.31510899961467,18.322674999999997 49.31502199961471,18.323169999999998 49.31453599961483,18.32337 49.31444699961844,18.325089 49.31330499961514,18.325616999999998 49.31342099961510 4,18.326475 49.31366299961
8	Prešovský kraj	POLYGON((19.883928 49.20417799964193,19.883958 49.20417799964193,19.887541999999996 49.20130299964263,19.887693999999996 49.20122699964265,19.88826 49.20094699964427,19.888584 49.20061199964278,19.889858 49.20056899964 28,19.8905339



3. zoraďte kraje podľa ich veľkosti (st_area). Veľkosť vypočítajte pomocou vhodnej funkcie a zobrazte v km^2 v SRID 4326.

```
SELECT name, ST_AREA(ST_ASTEXT(ST_TRANSFORM(pog.way, 4326))::geography)/1000000 as st_area
FROM planet_osm_polygon as pog WHERE admin_level='4' ORDER BY st_area DESC;
```

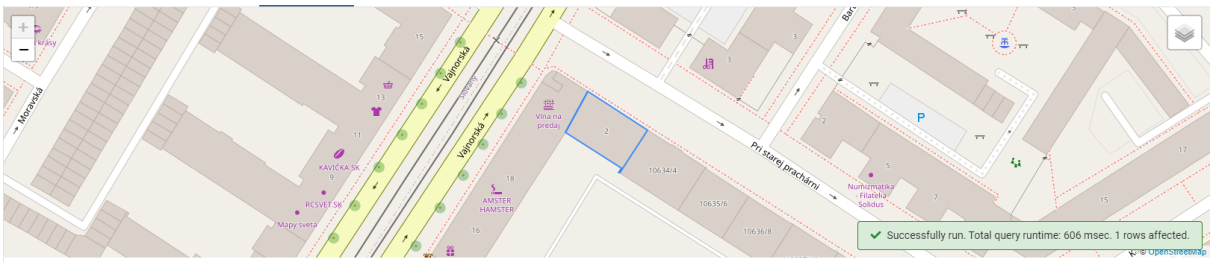
	name text	st_area double precision
1	Banskobystrický kraj	9454.543435236563
2	Prešovský kraj	8971.630508024704
3	Žilinský kraj	6806.915104947963
4	Košický kraj	6751.927549426834
5	Nitriansky kraj	6341.249779583104
6	Trenčiansky kraj	4501.797074560811
7	Trnavský kraj	4145.341158351734
8	Bratislavský kraj	2051.67405276842

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 domu som získal cez OSM - https://www.openstreetmap.org/way/118792392 konkrétne v sekcii uzly (dokopy 8 bodov)

```
INSERT INTO planet_osm_polygon(name, way)
VALUES ('My house', ST_TRANSFORM(ST_GeometryFromText
('LINESTRING(17.1309460 48.1603847,
 17.1309425 48.1603855,
 17.1309527 48.160396417,
 17.1307946 48.1604638,
 17.1307520 48.1604824,
 17.1308327 48.1605805,
 17.1310462 48.1604876,
 17.1309460 48.1603847)', 4326), 3857));
```

```
SELECT ST_TRANSFORM(way, 4326) FROM planet_osm_polygon WHERE name = 'My house';
```



5. zistite v akom kraji je váš dom.

Iba jednoduchý select kde som matchol kraj pomocou ST_intersects

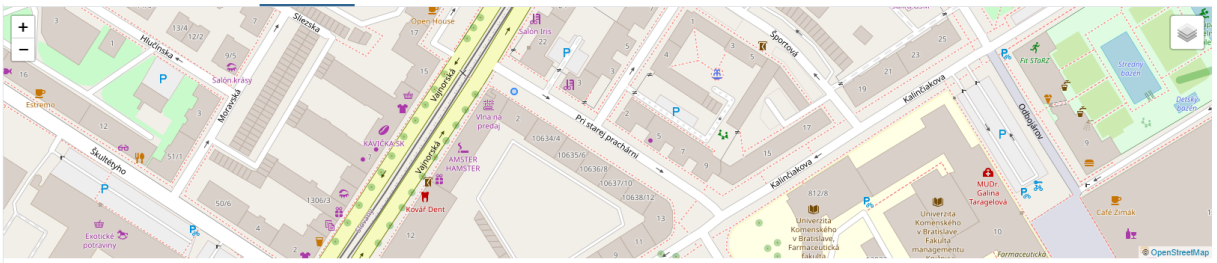
```
SELECT x.name, y.name
FROM planet_osm_polygon x, planet_osm_polygon y
WHERE x.name = 'My house' AND ST_INTERSECTS(x.way, y.way) AND y.admin_level = '4';
```

	name text	name text
1	My house	Bratislavský kraj

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

```
INSERT INTO planet_osm_point(name, way)
VALUES ('My location', ST_TRANSFORM(ST_GeometryFromText('POINT(17.130874350242697 48.16061599691293)', 4326), 3857));
```

```
SELECT ST_TRANSFORM(way, 4326) as loc FROM planet_osm_point WHERE name = 'My location';
```



Na mape už môžem vidieť že tá poloha úplne presná není keďže by som musel mať pc v strede ulice.

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

```
SELECT ST_CONTAINS(ST_TRANSFORM(x.way, 4326), ST_TRANSFORM(y.way, 4326))
FROM planet_osm_polygon x, planet_osm_point y
WHERE x.name = 'My house' AND y.name = 'My location';
```

	st_contains boolean
1	false

Výsledok vráti ako false pretože moja poloha (získaná cez google maps aj na PC a aj na mobile) nie je presná a tvári sa, že sedím vonku na ulici. Keby bola lokalita presná, výsledok by bol true.

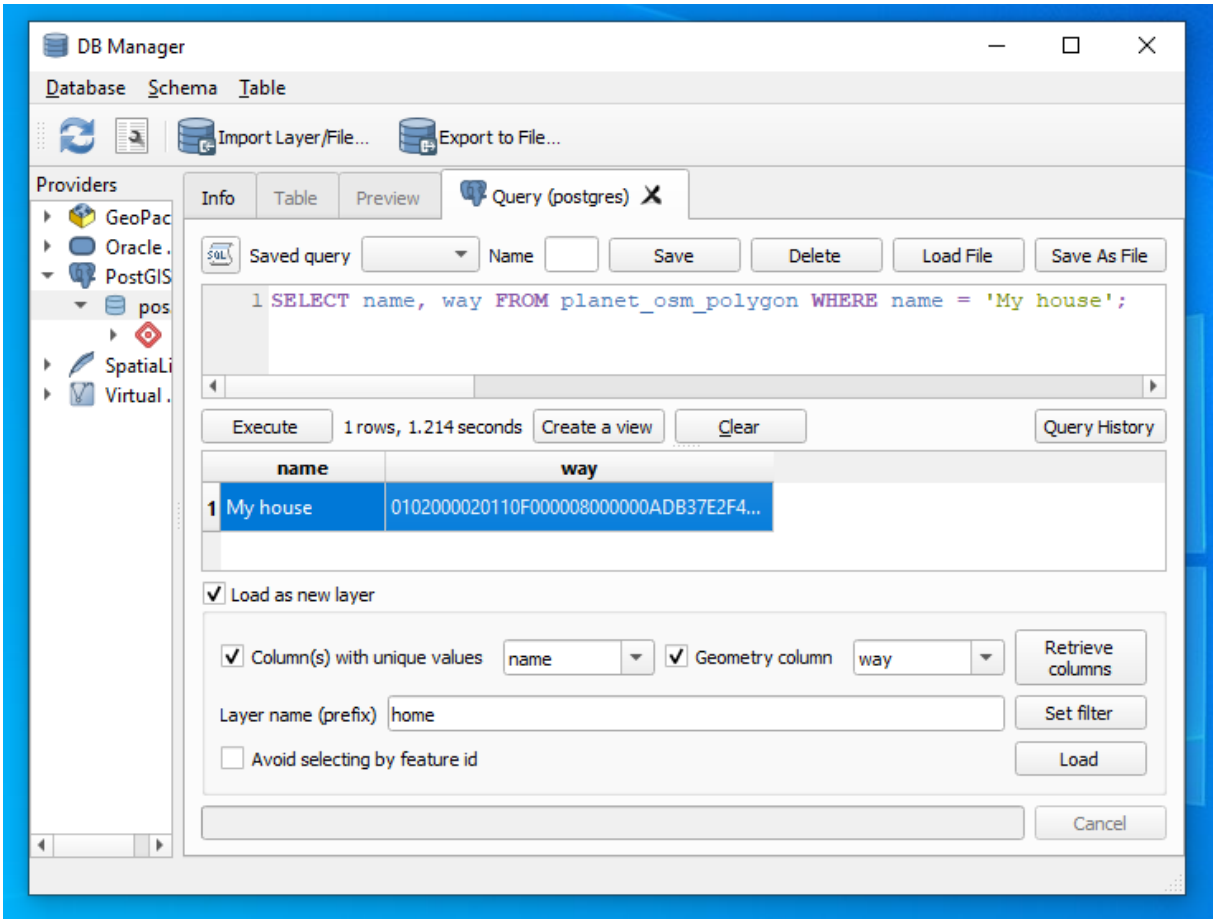
8. zistite ako ďaleko sa nachádzate od FIIT (name = 'Fakulta informatiky a informačných technológií STU'). Pozor na správny súradnicový systém – vzdialenosť musí byť skutočná.

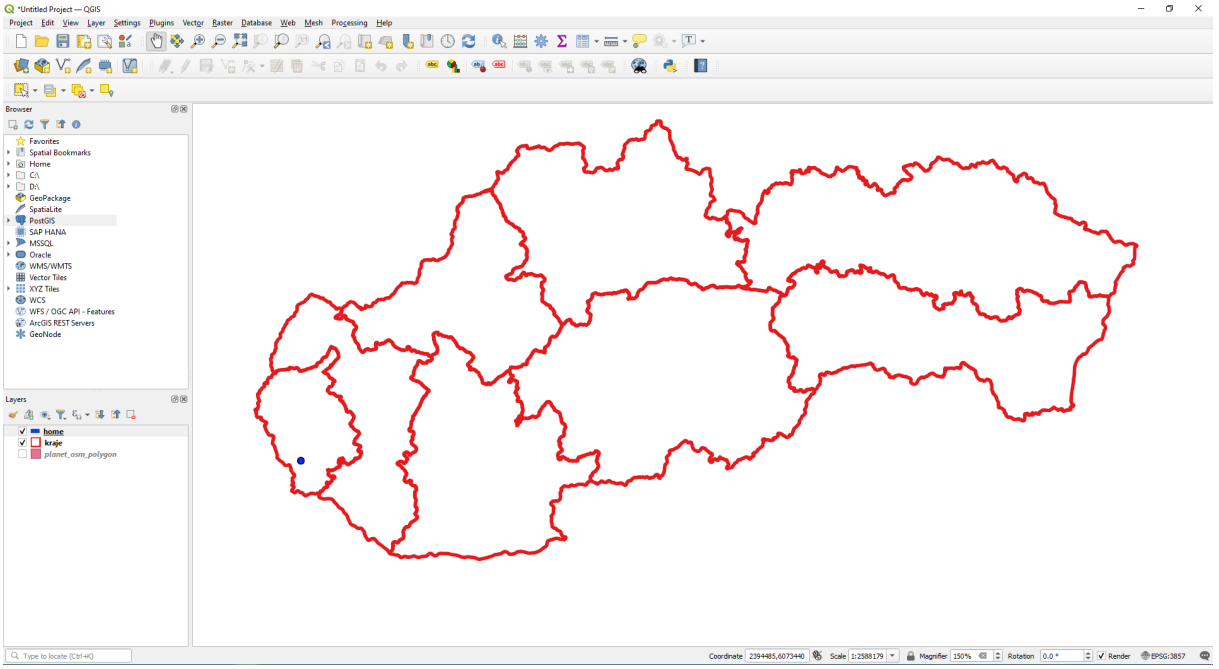
```
SELECT ST_DISTANCE(ST_TRANSFORM(x.way, 4326), ST_TRANSFORM(y.way, 4326), TRUE)/1000 as distance_in_km
FROM planet_osm_polygon x, planet_osm_point y
WHERE x.name = 'Fakulta informatiky a informačných technológií STU' AND y.name = 'My location';
```

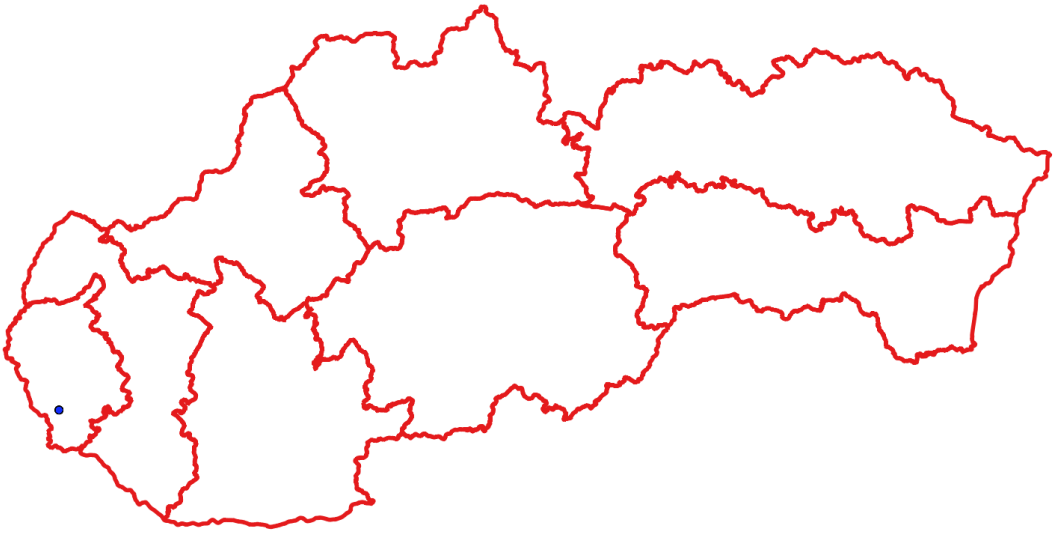
	distance_in_km double precision
1	4.4095420150099995

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

Po inštalácii QGIS som si vytvoril pripojenie na postgres databázu (resp. postgis), následne som prostredníctvom DBManager vytvoril SQL query pre kraje (úloha 1) a následne pre polygón reprezentujúci môj dom (úloha 4) a výsledky som pridal ako nový Layer. Kraje som vyznačil prostredníctvom červeného outline, môj dom pomocou modrého (na mape viditeľný ako modrá bodka).


DB Manager
Database Schema Table
Import Layer/File...
Export to File...
Providers
GeoPac
Oracle .
PostGIS
pos
SpatiaLi
Virtual .
Info
Table
Preview
Query (postgres)
Saved query
Name
Save
Delete
Load File
Save As File
1 SELECT name, way FROM planet_osm_polygon WHERE name = 'My house';
Execute
1 rows, 1.214 seconds
Create a view
Clear
Query History
name
way
1 My house
0102000020110F000008000000ADB37E2F4...
Load as new layer
Column(s) with unique values
name
Geometry column
way
Retrieve columns
Layer name (prefix)
home
Set filter
Avoid selecting by feature id
Load
Cancel


*Untitled Project — QGIS
Project Edit View Layer Settings Plugins Vector Raster Database Web Mesh Processing Help
Browser
Favorites
Spatial Bookmarks
Home
C:\
D:\
GeoPackage
SpatiaLite
PostGIS
SAP HANA
MSSQL
Oracle
WMS/WMTS
Vector Tiles
XYZ Tiles
WCS
WFS / OGC API - Features
ArcGIS REST Servers
GeoNode
Layers
home
kraje
planet_osm_polygon
Type to locate (Ctrl+K)
Coordinate 2394485,6073440
Scale 1:2588179
Magnifier 150%
Rotation 0.0 °
Render
EPSG:3857



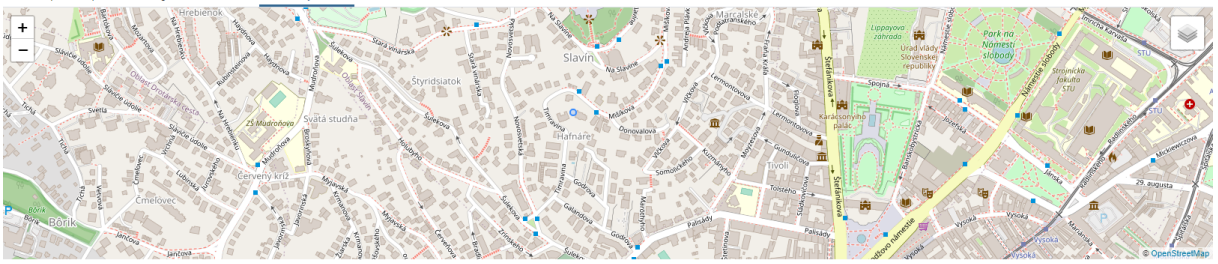
10. Zistite súradnice centroidu (ťažiska) plošne najmenšieho okresu (vo výsledku nezabudnite uviesť aj EPSG kód súradnicového systému).

Nevedel som či hľadať okres v BA, na celom Slovensku alebo v celých dátach ale nakoniec to vyšlo v BA čiže všetko ok

```
SELECT name, ST_X(ST_TRANSFORM(ST_CENTROID(way), 4326)) as centroid_long, ST_Y(ST_TRANSFORM(ST_CENTROID(way), 4326)) as centroid_lat,
ST_TRANSFORM(ST_CENTROID(way), 4326) as way ,
ST_AREA(ST_ASTEXT(ST_TRANSFORM(way, 4326))::geography)/1000000 as st_area, ST_SRID(ST_CENTROID(ST_TRANSFORM(way, 4326)))
FROM planet_osm_polygon WHERE name ilike '%okres %' ORDER BY st_area ASC LIMIT 1;
```

	name text	centroid_long double precision	centroid_lat double precision	way geometry	st_area double precision	st_srid integer
1	okres Bratislava I	17.099408518274394	48.150981842565855	0101000020E6100000AEEE2ED6721931401A0E7E5F53134840	9.587809061799536	4326

Data Output Explain Messages Notifications Geometry Viewer

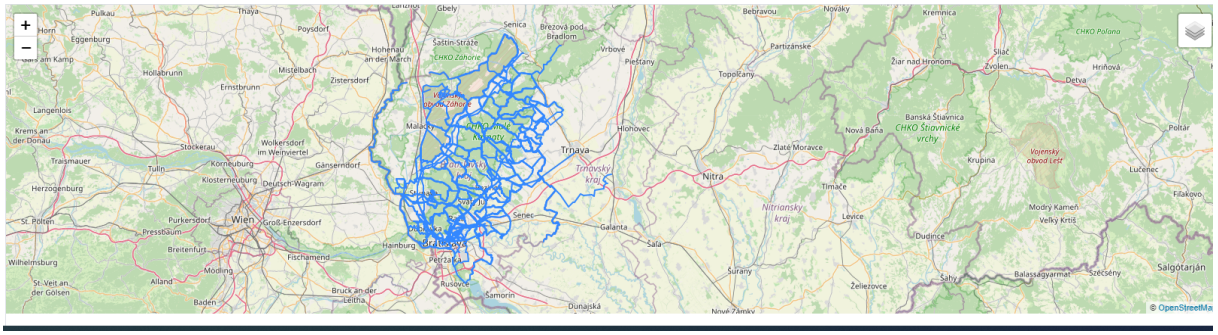


11. Vytvorte priestorovú tabuľku všetkých úsekov ciest, ktorých vzdialenosť od vzájomnej hranice okresov Malacky a Pezinok je menšia ako 10 km.

```
SELECT ST_TRANSFORM(way, 4326) as roads
FROM planet_osm_roads
WHERE ST_DISTANCE (
    (SELECT ST_TRANSFORM(ST_INTERSECTION(p1.way, p2.way), 4326)
    FROM planet_osm_polygon p1, planet_osm_polygon p2
    WHERE p1.name = 'okres Pezinok' AND p2.name = 'okres Malacky'), ST_TRANSFORM(way, 4326), TRUE
)/1000 < 10;
```

Data Output Explain Messages Notifications Geometry Viewer

	roads geometry
1	0102000020E61000001A000000BD923F7331053140539941 30FD294840DF0F898C58053140464 8C2CDF3294840A40F02976C0531407E9C8722EE294840DB26817280053140C542E2D2E7294840F828C8748D0531402E90C435E32948403DBDAD4F94053140BC0976C1E0294840805193
2	0102000020E6100000030000005 4C90050C5053140CDC90A43E429484022CFE451E00531 40DCB28F6EDF2948408 0FA230C03063140AA6A077CD9294840
3	0102000020E61000002A00000054C90050C5053140CDC90A43E4294840D57EC6E0BC053140609716A9E6294840BB9DD89EB40531404920B598E92948401299485AF0531406DDBA923EC2948407899 61A3AC053140C434E0F1ED2948407F06E5C7A9053140 87FC51E3EF294840F299E
4	0102000020E61000001E00000040B6D1A524063140BA75E1163D2A48409F11B6E919063140 4AADAD9D282A4840534C9473180631401A16ED4A262A4840EE1692DD16063140584E7B59242A4840C5837703160631402985F65232A4840E3E4D9401506314007FA3381222A4840096B08
5	0102000020E61000000200000CF95AD9A8C05314071AC4E82ED29484054C90050C5053140CDC90A43E4294840
6	0102000020E61000001E00000014DF9A85870631407657C464602A48400CA77114C5063140ECA769CB6F2A484095325F14F3063140489DE4677E2A484089D2DEE00B0731407A47FC998B2A4840E7502BF1250731402298728 3A12A4840C10644882B0731407E6546F3B62A4840EE75F7
7	0102000020E61000002E0500008F1CE90C8CF8304018F74FA6D1444840B81798158AF83040AD405D13D2444840137B681F2BF8304060D75FE5D044484042D2A755F4F7304009C0A813D0444840DE0033DFC1F730408C9EC456D04448409CFBABC77DF730403353C3EDD0444840F8E3F
8	0102000020E61000000200000B0FA230C03063140AA6A077CD9294840EE456A903806314 0FF82D446D0294840
9	0102000020E61000000200000EE456A903806314 0FF82D446D0294840 7F947DFC4A0631404D83C144CD294840
10	0102000020E6100000020000007F947DFC4A0631404D83C144CD294840463E54BF7706314089C95134C5294840
11	0102000020E61000000C000000463E54BF7706314089C95134C5294840F819170E840631409BBE79CAC52948400C7217618A063140E49D408CC6294840E6DF7888900631 40A41E448BC729484093324EE89A063140EFA09D87C9294840BF7630629F0631402B719767CA2948404676A5E
12	0102000020E61000000200000463E54BF7706314089C95134C52948406979799510073140 2E7F00BCAA294840
13	0102000020E61000001800000002011AEDBFB053140F16244566D2948400C6866EF607063140BC00B0A48129484023DF5B470B0631400DBAF344987294840E1CFF0660D0631406D07771B8A2948402 16B5732110631407 2A388658E2948401882D20616063140 1C0E3E9292294840074037 8
14	0102000020E61000000700000069797995100731402E7F00BCAA29484060F01FC14307314021EBB8DFA1294840DB673AE05F073140675DC9F89C294840F57C72B966073140EB6F2FD39B294840C79D2D7B6D073140715483899A2948408306 51AD74073140CB8B171A992948402AC4C8



12. Jedným dopytom zistite 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.

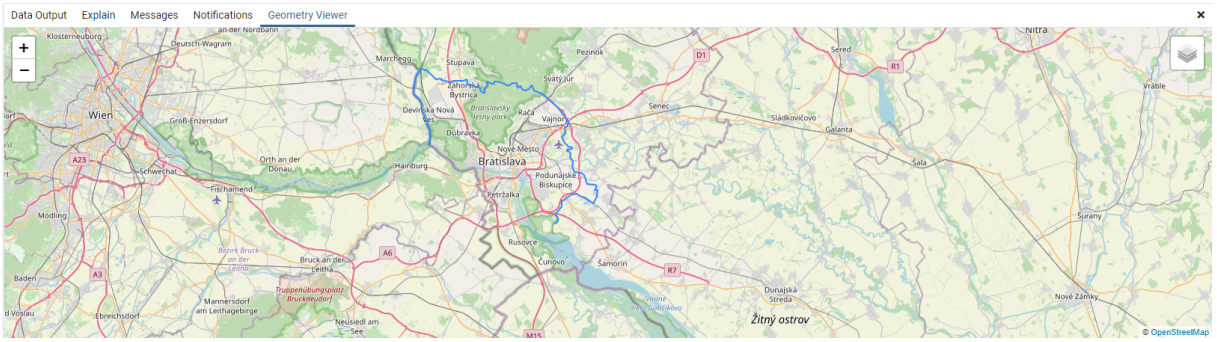
Dáta vo formáte .gpkg som do db dostal pomocou ogr2ogr z GDAL.

```
AzureAD+JurajBedej@DESKTOP-SAROJV7 MINGW64 ~/Desktop/PDT/PDT-3/PDT-3/data (main)
$ ogr2ogr -f PostgreSQL "PG:user=postgres password=postgres dbname=gis host=loca
lhost port=5433" USJ_hranice_0.gpkg -lco LAUNDER=NO
```

```
SELECT k."NM5", k."IDN5", ST_TRANSFORM(way, 4326) FROM ku_0 as k, planet_osm_roads
WHERE k."NM3"='Bratislava III' AND ST_INTERSECTS(ST_TRANSFORM(k."Shape", 4326), ST_TRANSFORM(way, 4326))
ORDER BY ST_LENGTH(ST_TRANSFORM(way, 4326), TRUE) DESC
LIMIT 1;
```

Data Output | Explain | Messages | Notifications | Geometry Viewer

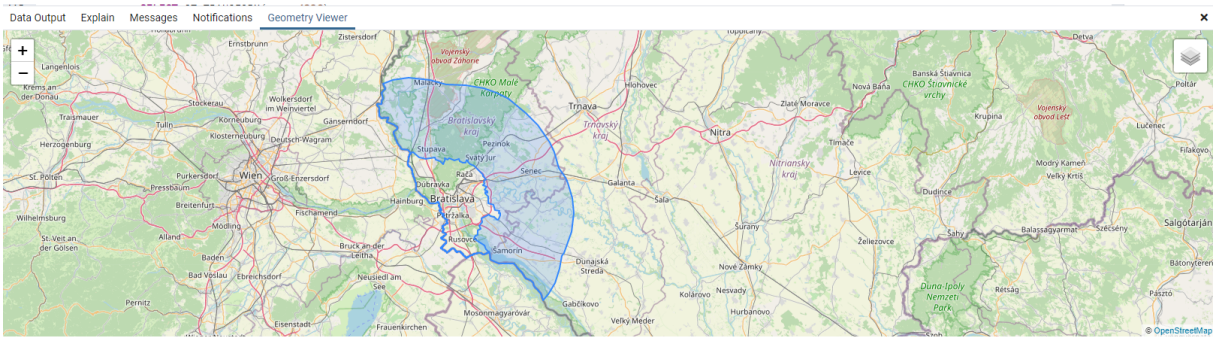
	NM5 character varying (50)	IDN5 integer	st_transform geometry
1	Vajnory	805700	0102000020E6100000010600004081 77F2E9F930407CBD12190C16484062BE7273E0F9304050FC4B231C164840E5F1B4FCC0F93040206996E656164840BF9657AEB7F930406DF1C0E270164840722D5A80B6F9304099F9C1DB8316484036936FB6B9



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.

```
WITH Okolie_Bratislavy
AS
(
    SELECT ST_INTERSECTION(
        (SELECT ST_TRANSFORM(way, 4326)
        FROM planet_osm_polygon
        WHERE name = 'Slovensko')
        ,
        ST_DIFFERENCE(
            (SELECT ST_BUFFER (
            (SELECT ST_UNION(sub.way)
            FROM (
                SELECT ST_TRANSFORM(way, 4326) as way
                FROM planet_osm_polygon
                WHERE name = 'Bratislava'
            ) as sub)::geography, 20000 )::geometry)
            ,
            (SELECT ST_UNION(sub.way)
            FROM (
                SELECT ST_TRANSFORM(way, 4326) as way
                FROM planet_osm_polygon
                WHERE name = 'Bratislava'
            ) as sub)
        )
    ) as way
)
SELECT st_area(way, TRUE)/1000^2, way FROM Okolie_Bratislavy;
```

	?column? double precision	way geometry
1	1479.101214875604	0107000020E6100000930300000103000000001000000CF080000958098840BF530408E3E66D7BD214840111D024702F5304014C7C8C9C4214840211ADD41ECF43040A35288B7CE214840D13E56F0DBF430405793D138D4214840E199D024B1F4304098C411DBDD



Pomocou ST_UNION som si vytváral polygón ktorý obsahoval celú Bratislavu (keďže neviem či v dátach taký už bol, ale napr. s name Bratislava tam bolo polygónov hneď niekoľko tak som ich iba mergol do jedného s ktorým som pracoval pri ST_BUFFER a ST_DIFFERENCE). Tiež mi tam ostala tá hranica s Rakúskom ale na Slacku bolo písané že to nevadí tak som to ďalej neriešil.