

# Pokročilé databázové technológie

Zadanie 3 – PostGIS

Marek Adamovič

## Obsah

1. otázka.....	3
2. otázka.....	4
3. otázka.....	5
4. otázka.....	6
5. otázka.....	7
6. otázka.....	8
7. otázka.....	9
8. otázka.....	10
9. otázka.....	11
10. otázka .....	12
11. otázka .....	13
12. otázka .....	15
13. otázka .....	16

## 1. otázka

### Otázka:

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

### Odpoveď:

Pre import som použil osm2pgsql. Cez application stack builder som si pridal postgresu do postgresu.

```
1 CREATE EXTENSION IF NOT EXISTS postgis;
```

```
D:\postgress\osm2pgsql>osm2pgsql -c -d maps -U postgres --password -H localhost -S D:\postgress\osm2pgsql\default.style D:\skola2022_2023\PD1\zadanie3\slovakia-latest.osm.pbf
2022-10-28 17:21:31 osm2pgsql version 1.7.1
Password:
2022-10-28 17:21:33 Database version: 14.5
2022-10-28 17:21:33 PostGIS version: 3.3
2022-10-28 17:21:33 Setting up table 'planet_osm_point'
2022-10-28 17:21:33 Setting up table 'planet_osm_line'
2022-10-28 17:21:33 Setting up table 'planet_osm_polygon'
2022-10-28 17:21:33 Setting up table 'planet_osm_roads'
2022-10-28 17:22:37 Reading input files done in 64s (1m 4s).
2022-10-28 17:22:37 Processed 27363095 nodes in 4s - 6841k/s
2022-10-28 17:22:37 Processed 3569474 ways in 46s - 78k/s
2022-10-28 17:22:37 Processed 45820 relations in 14s - 3k/s
2022-10-28 17:22:39 Clustering table 'planet_osm_roads' by geometry...
2022-10-28 17:22:39 Clustering table 'planet_osm_point' by geometry...
2022-10-28 17:22:39 Clustering table 'planet_osm_polygon' by geometry...
2022-10-28 17:22:39 Clustering table 'planet_osm_line' by geometry...
2022-10-28 17:23:47 Creating geometry index on table 'planet_osm_roads'...
2022-10-28 17:23:50 Analyzing table 'planet_osm_roads'...
2022-10-28 17:23:52 Creating geometry index on table 'planet_osm_point'...
2022-10-28 17:24:14 Analyzing table 'planet_osm_point'...
2022-10-28 17:24:14 All postprocessing on table 'planet_osm_point' done in 95s (1m 35s).
2022-10-28 17:24:55 Creating geometry index on table 'planet_osm_line'...
2022-10-28 17:25:17 Analyzing table 'planet_osm_line'...
2022-10-28 17:25:17 All postprocessing on table 'planet_osm_line' done in 158s (2m 38s).
2022-10-28 17:25:39 Creating geometry index on table 'planet_osm_polygon'...
2022-10-28 17:26:13 Analyzing table 'planet_osm_polygon'...
2022-10-28 17:26:14 All postprocessing on table 'planet_osm_polygon' done in 214s (3m 34s).
2022-10-28 17:26:14 All postprocessing on table 'planet_osm_roads' done in 73s (1m 13s).
2022-10-28 17:26:14 osm2pgsql took 281s (4m 41s) overall.
```

## 2. otázka

### Otázka:

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

### Odpoveď:

Súradnicový systém 4326 využíva longitude a latitude. Z toho dôvodu sme si museli transformovať dáta z 3857 do 4326 súradnicového systému.

```
1 SELECT
2     name,
3     ST_ASTEXT(ST_TRANSFORM(way, 4326)) latitude_longitude_polygon
4 FROM planet_osm_polygon
5 WHERE admin_level = '4';
```

	name text	latitude_longitude_polygon text
1	Trenčiansky kraj	POLYGON((17.353035699999996 48.7775265997475,17.353104 48.77746609974752,17.3539395 48.77677139974771,1...
2	Trnavský kraj	POLYGON((16.933595 48.600625999791326,16.933598999999997 48.60019099979144,16.933632999999997 48.59973...
3	Nitriansky kraj	POLYGON((17.705419 47.75898499999446,17.705507699999995 47.75889909999449,17.707337999999996 47.757359...
4	Žilinský kraj	POLYGON((18.322605 49.31510899961467,18.322674999999997 49.31502199961471,18.323169999999998 49.314535...
5	Prešovský kraj	POLYGON((19.883928 49.20417799964193,19.883958 49.20417799964193,19.887541999999996 49.20130299964263,1...
6	Bratislavský kraj	POLYGON((16.833189099999995 48.380524099845566,16.8332193 48.380208499845644,16.833291399999997 48.379...
7	Košický kraj	POLYGON((20.181221999999998 48.74099699975657,20.181416 48.74054899975667,20.181695999999995 48.739917...
8	Banskobystrický kraj	POLYGON((18.478672999999997 48.54860199980419,18.478683 48.54857799980419,18.479153 48.548347999804236,...

### 3. otázka

#### Otázka:

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

#### Odpoveď:

Pre vypísanie správnej hodnoty sme museli pretypovať typ dát z geometrických na geografické, o čo sa nám postará samotná funkcia ST\_AREA, keď do nej pridáme tretí argument (ktorý hovorí o použití spheroidu). Následne sme dostali výsledok v metroch štvorcových, ktorý sme prerátali na kilometre štvorcové.

```
1 SELECT
2     name,
3     ST_AREA(ST_TRANSFORM(way,4326), true) / 1000000 km_squared
4 FROM planet_osm_polygon
5 WHERE admin_level = '4'
6 ORDER BY km_squared
```

Data output Messages Notifications

	name text	km_squared double precision
1	Bratislavský kraj	2051.660249334432
2	Trnavský kraj	4145.350058274841
3	Trenčiansky kraj	4501.795536203455
4	Nitriansky kraj	6341.2406677586605
5	Košický kraj	6751.964233845743
6	Žilinský kraj	6806.886696700907
7	Prešovský kraj	8971.623953642224
8	Banskobystrický kraj	9454.54638091422

### 4. otázka

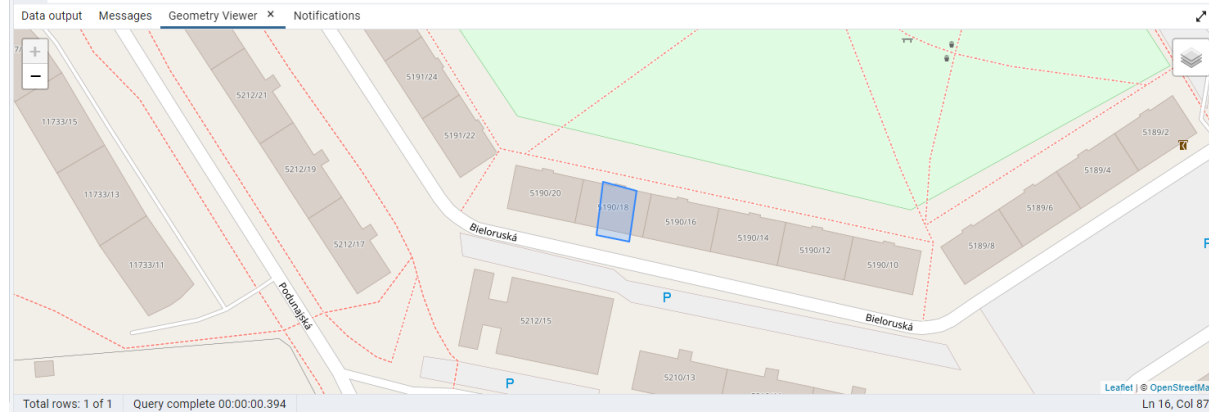
#### Otázka:

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.

#### Odpoveď:

Polygon musíme vytvoriť v súradnicovom systéme 4326 (lebo ho vytvárame pomocou latitude a longitude), avšak do tabuľky ho vkladáme v systéme 3857 (keďže máme tak zadefinovanú tabuľku).

```
1 INSERT INTO planet_osm_polygon(name, admin_level, way)
2 VALUES(
3     'moje_byvanie',
4     -1,
5     ST_TRANSFORM(ST_POLYGONFROMTEXT(
6         'POLYGON((
7             17.198481570188463 48.13911843613888,
8             17.19850320540223 48.13922604291423,
9             17.19861325011859 48.139206353721775,
10            17.19858852992868 48.13909620053269,
11            17.198481570188463 48.13911843613888))',
12            4326
13        ), 3857)
14 )
15
16 SELECT name, ST_TRANSFORM(way, 4326) FROM planet_osm_polygon WHERE admin_level = '-1';
```



## 5. otázka

### Otázka:

Zistíte v akom kraji je váš dom.

### Odpoveď:

Použili sme funkciu ST\_WITHIN, ktorá zisťuje, či sa celý jeden objekt (v tomto prípade môj byt) nachádza v druhom (v tomto prípade kraje Slovenska).

```
1 SELECT * FROM(  
2     SELECT  
3         name,  
4         ST_WITHIN(  
5             (SELECT way FROM planet_osm_polygon WHERE admin_level = '-1'),  
6             way  
7         ) my_house_is_there  
8     FROM planet_osm_polygon  
9     WHERE admin_level = '4'  
10 ) q1  
11 WHERE my_house_is_there
```

Data output Messages Notifications

	name text	my_house_is_there boolean
1	Bratislavský kraj	true

## 6. otázka

### Otázka:

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

### Odpoveď:

Podobne ako v úlohe č.4 aktuálnu polohu musíme vytvoriť v súradnicovom systéme 4326 (lebo ho vytvárame pomocou latitude a longitude), avšak do tabuľky ho vkladáme v systéme 3857 (keďže máme tak zadefinovanú tabuľku).

```
1 INSERT INTO planet_osm_point(name, admin_level, way)
2 VALUES(
3     'moja_pozicia',
4     -1,
5     ST_TRANSFORM(ST_POINTFROMTEXT('POINT(17.19854902450499 48.13916593736717)', 4326), 3857)
6 )
7
8 SELECT name, admin_level, ST_TRANSFORM(way, 4326) FROM planet_osm_point WHERE admin_level = '-1';
```

Total rows: 1 of 1 Query complete 00:00:00.207



## 7. otázka

### Otázka:

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

### Odpoveď:

Použil som rovnaký spôsob riešenia ako v úlohe č.5, teda pomocou funkcie ST\_WITHIN.

```
1 SELECT
2     ST_WITHIN(
3         (SELECT way FROM planet_osm_point WHERE name = 'moja_pozicia'),
4         (SELECT way FROM planet_osm_polygon WHERE name = 'moje_byvanie')
5     ) som_doma
```

Data output Messages Notifications

≡+ 📄 ▼ 📋 🗑️ 🗄️ ⬇️ 📈

	som_doma boolean 🔒
1	true

### 8. otázka

#### Otázka:

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

#### Odpoveď:

Vo funkcii ST\_DISTANCE sme použili aj tretí argument (ktorý nám hovorí o použití spheroidu na vypočítanie vzdialenosti), vďaka čomu funkcia chápe prvé dva parametre ako typ geography a nie geometry.

1	SELECT
2	ST_DISTANCE(
3	(SELECT ST_TRANSFORM(way, 4326) FROM planet_osm_point WHERE name = 'moja_pozicia'),
4	(SELECT ST_TRANSFORM(way, 4326) FROM planet_osm_polygon WHERE name = 'Fakulta informatiky a informačných technológií STU'),
5	true
6	)/1000 vzdialenost_od_skoly_v_km

Data output	Messages	Notifications
<div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div>		
vzdialenost_od_skoly_v_km		
double precision		
1	9.51507326225	

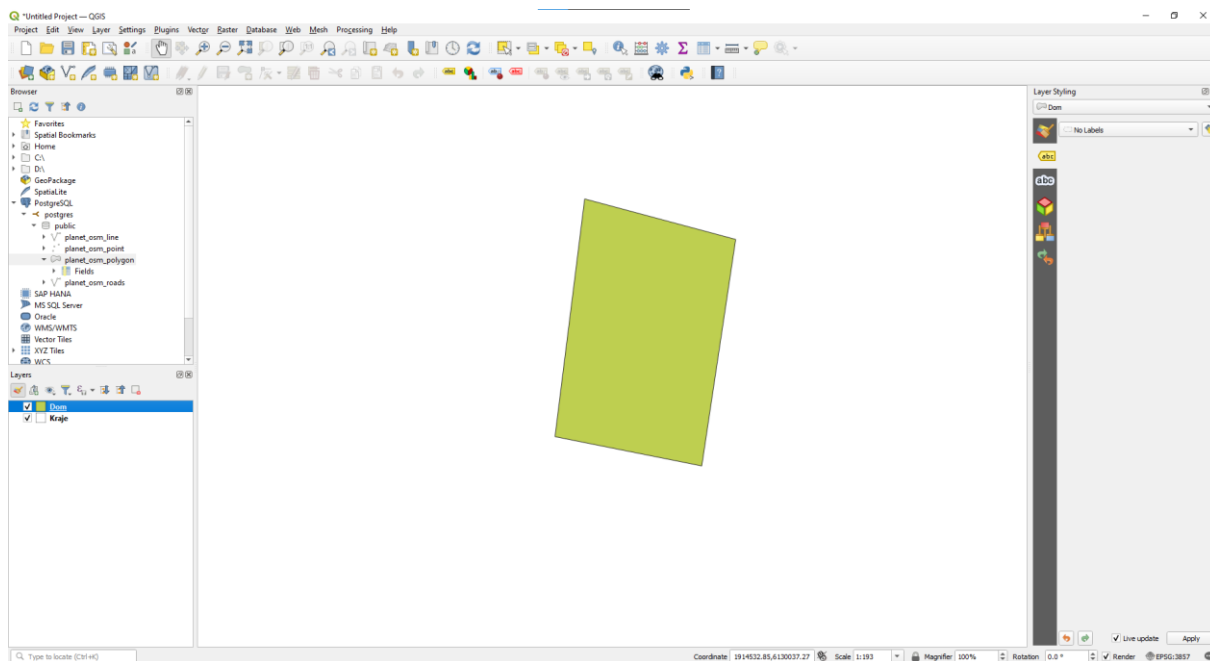
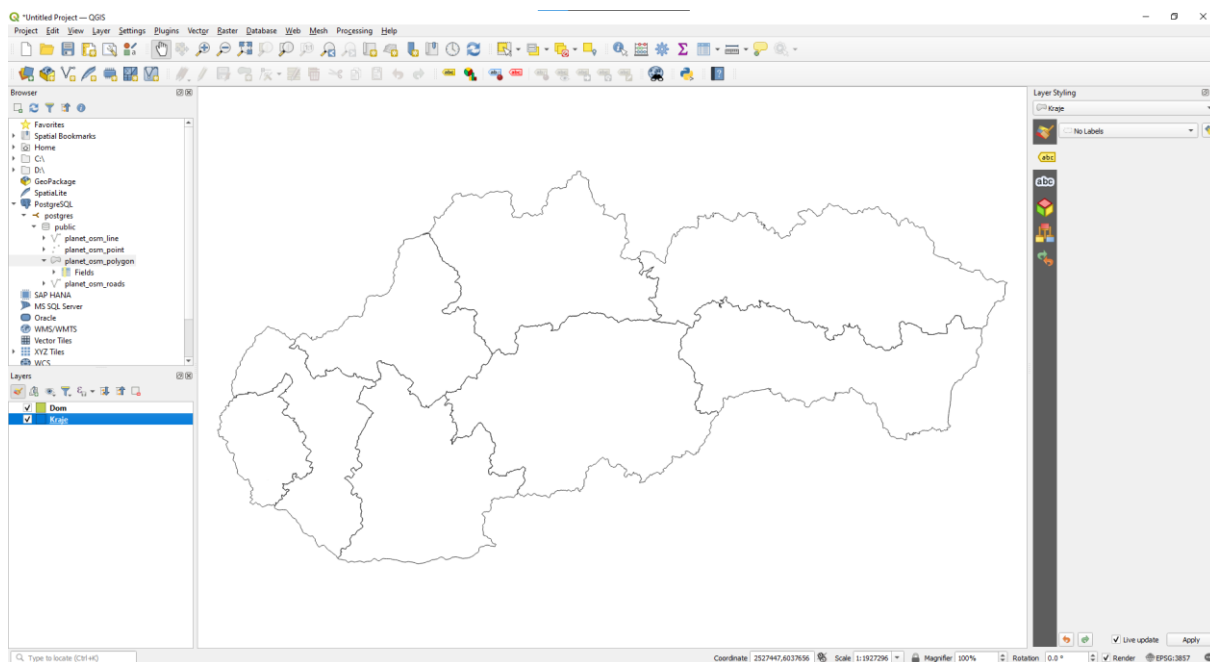
### 9. otázka

#### Otázka:

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

#### Odpoveď:

Keďže objekt domu je oveľa menší ako kraje Slovenska, nie je ho vidno na prvom obrázku. Avšak keď klikneme na vrstvu s domom a dáme „zoom to layer“, tak sa vieme naň ľahko dostať.



## 10. otázka

### Otázka:

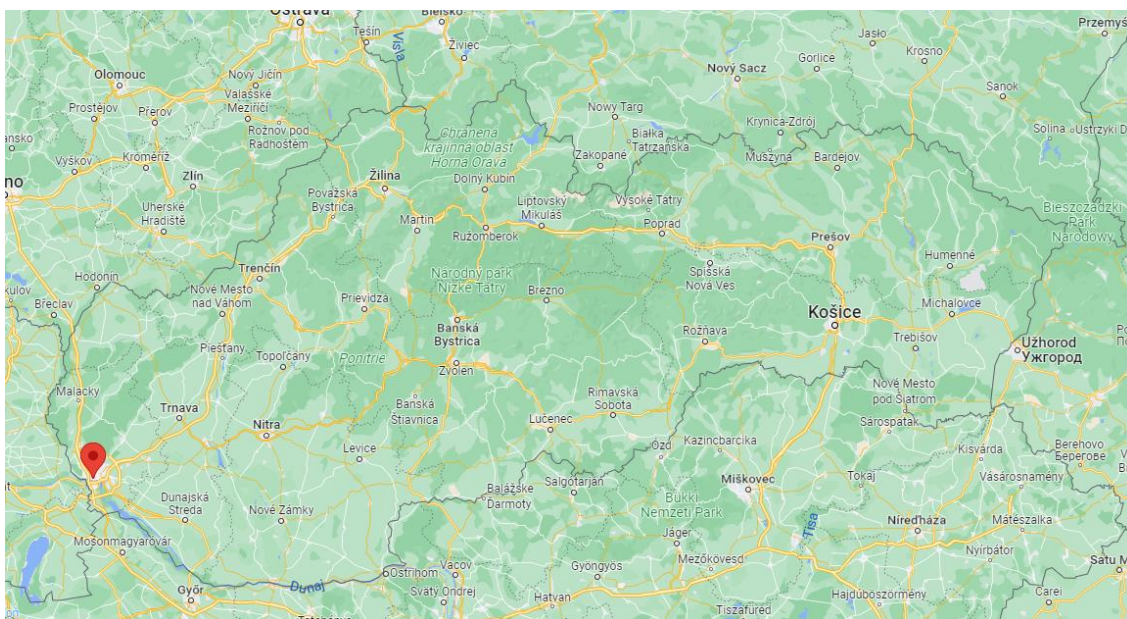
Zistite súradnice centroidu (ťažiska) plošne najmenšieho okresu (vo výsledku nezabudnite uviesť aj EPSG kód súradnicového systému).

### Odpoveď:

Výsledok máme v súradnicovom systéme EPSG-4326, teda je vyjadrený pomocou latitude a longitude. Na druhom obrázku sa nachádza screenshot z google máp, kde sme zadali danú latitude a longitude.

```
1 SELECT
2     name,
3     ST_AREA(ST_TRANSFORM(way, 4326), true) / 1000000 km_squared,
4     ST_X(ST_CENTROID(ST_TRANSFORM(way, 4326))) latitude,
5     ST_Y(ST_CENTROID(ST_TRANSFORM(way, 4326))) longitude,
6     4326 EPSG
7 FROM planet_osm_polygon
8 WHERE name LIKE 'okres%'
9 ORDER BY km_squared
10 LIMIT (1)
```

	name text	km_squared double precision	latitude double precision	longitude double precision	epsg integer
1	okres Bratislava I	9.588613370524051	17.0994108837330	48.1509790762363	4326



## 11. otázka

### Otázka:

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

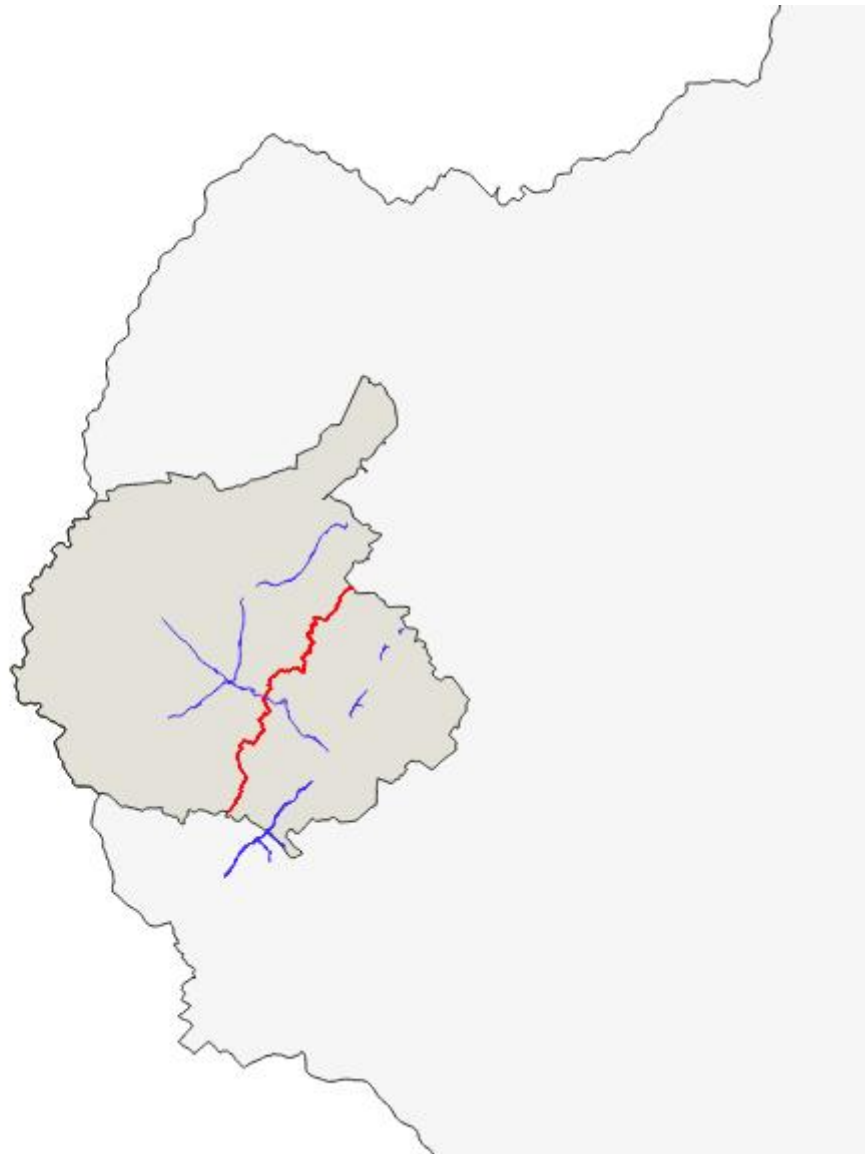
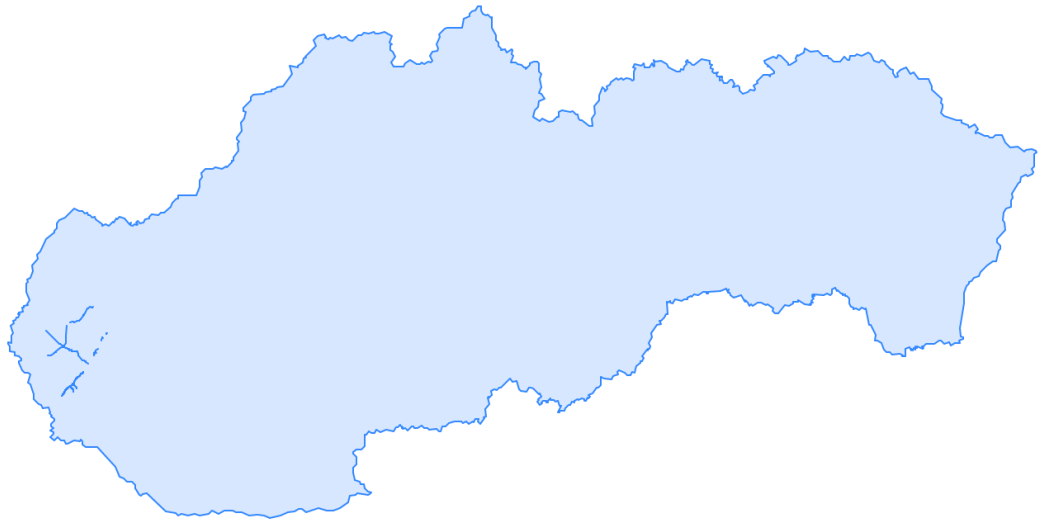
### Odpoveď:

Najskôr sme si izolovali hranicu medzi okresom Malacky a okresom Pezinok pomocou funkcie ST\_INTERSECTION. Následne sme počítali veľkosť najmenších vzdialeností medzi všetkými cestami (cesty sme z tabuľky vyfiltrovali pomocou WHERE) na Slovensku a touto hranicou (pomocou funkcie ST\_DISTANCE). Vo vonkajšom dopyte sme vyfiltrovali len tie cesty, ktoré sú bližšie ako 10 kilometrov od hranice okresov. Celý dopyt sme využili v klauzule CREATE TABLE AS, ktorá nám vytvorila tabuľku s cestami, ktoré spĺňali podmienku. Na druhom obrázku môžeme vidieť vyobrazenie ciest spolu s celým Slovenskom. Na treťom obrázku zas vidíme vyobrazenie ciest spolu s okresom Malacky a okresom Pezinok (kde hranica okresov je vyznačená červenou a cesty modrou farbou).

```
1 CREATE TABLE IF NOT EXISTS malacky_pezinok_roads AS
2 (SELECT *
3 FROM(
4     SELECT
5         ST_DISTANCE(
6             ST_INTERSECTION(
7                 (
8                     SELECT DISTINCT ON (name, boundary) way
9                     FROM planet_osm_polygon
10                    WHERE name = 'okres Malacky' AND boundary = 'administrative'
11                ),
12                (
13                    SELECT DISTINCT ON (name, boundary) way
14                    FROM planet_osm_polygon
15                    WHERE name = 'okres Pezinok' AND boundary = 'administrative'
16                )
17            ), roads_table.way
18        ) / 1000 AS distance_in_km,
19        roads_table.*
20     FROM planet_osm_roads roads_table
21     WHERE (roads_table.motorcar IS NOT NULL OR roads_table.highway IS NOT NULL) AND roads_table.railway IS NULL
22 ) inter_subset
23 WHERE distance_in_km < 10)
```

```
1 SELECT way FROM malacky_pezinok_roads
2 UNION
3 SELECT way FROM (SELECT ST_UNION(way) way FROM planet_osm_polygon WHERE admin_level = '4' GROUP BY admin_level) slovensko;
```

Data output Messages Geometry Viewer X Notifications



## 12. otázka

### Otázka:

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

### Odpoveď:

Najskôr sme vytiahli dáta z tabuľky `osm_polygon`, avšak len polygón s menom nášho okresu (Bratislava II). Túto informáciu sme pomocou `JOIN` spojili s cestami, ktoré pretínajú daný okres. Tieto pretínajúce cesty sme zlúčili do jednej pomocou `ST_UNION` a `GROUP BY`. Taktiež sme k týmto informáciami pridali pomocou `JOIN` jednotlivé katastrálne územia, ktoré patria do okresu Bratislava II. Medzi jednotlivými katastrálnymi územiami („Shape“) a spojenými cestami sme vykonali prienik a vypočítali sme dĺžku územia ciest na konkrétnych katastrálnych územiach. V poslednom kroku stačilo zostupne zoradiť katastrálne územia pomocou vypočítanej dĺžky a vybrať prvý záznam, čo bol Ružinov.

```
1 SELECT
2     idn5,
3     nm5,
4     ST_LENGTH(
5         ST_INTERSECTION(
6             ST_TRANSFORM(ST_UNION(ST_INTERSECTION(pop.way, por.way)), 4326),
7             ST_TRANSFORM("Shape", 4326)
8         )
9     ) road_length
10 FROM planet_osm_polygon pop
11 JOIN planet_osm_roads por ON ST_INTERSECTS(pop.way, por.way)
12 JOIN ku_0 kat ON kat.lau1_code = 'SK0102'
13 WHERE pop.name = 'okres Bratislava II'
14 GROUP BY pop.name, kat.nm5, kat.idn5, kat."Shape"
15 ORDER BY road_length DESC LIMIT 1
```

Data output Messages Geometry Viewer X Notifications

	idn5 integer	nm5 character varying (50)	road_length double precision
1	805556	Ružinov	1.8830567340326847

## 13. otázka

### Otázka:

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

### Odpoveď:

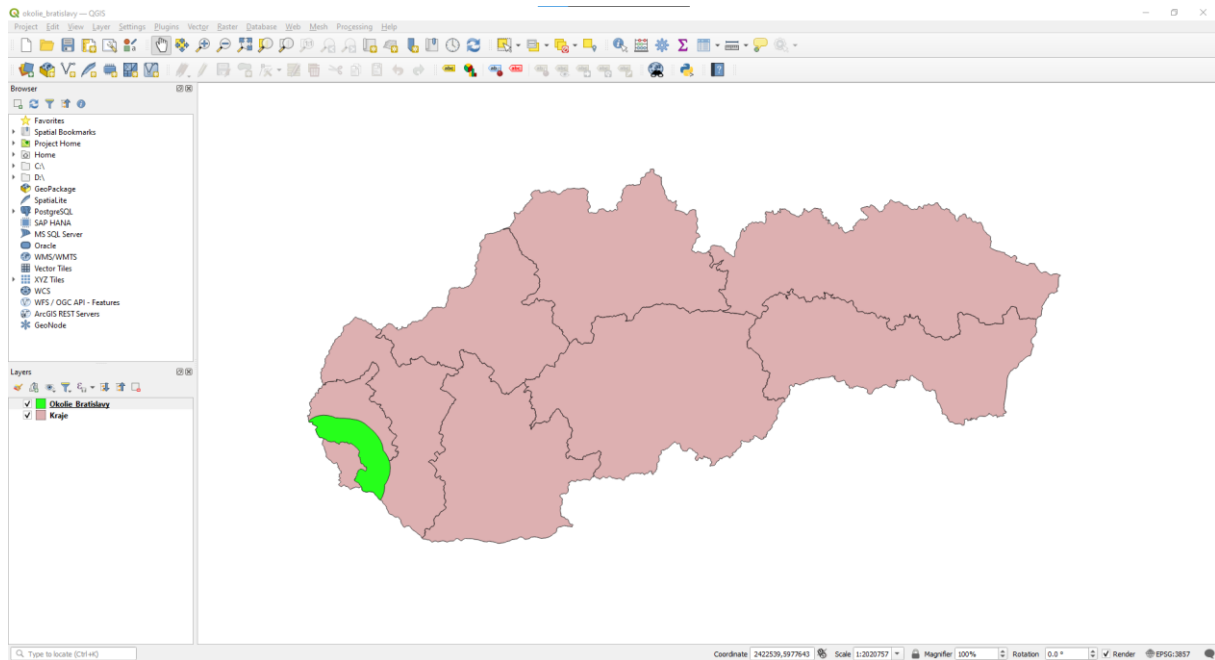
Najskôr sme si vytvorili subset (využijeme ho neskôr v dopyte) pomocou WITH, ktorý obsahuje celé Slovensko (ktoré sme pospájali pomocou krajov) a Bratislavu. Následne sme vytvorili okolo Bratislavy kruh s polomerom 20 kilometrov (s funkciou ST\_BUFFER). Ďalší krok bol vytvoriť prienik tohto kruhu a Slovenska (pomocou funkcie ST\_INTERSECTION) a následné odstránenie plochy samotnej Bratislavy (pomocou funkcie ST\_DIFFERENCE). Takto vytvorenú plochu sme vložili do tabuľky a následne importovali spolu s kraji do programu QGIS (pre lepšie zobrazenie). Na záver sme zbehli dopyt, ktorý nám vypočítal plochu v kilometroch štvorcových danej plochy (885.87km<sup>2</sup>).

```
1 WITH subset AS(
2     SELECT
3         (
4             SELECT way
5             FROM planet_osm_polygon
6             WHERE name = 'Bratislava' AND boundary = 'administrative'
7         ) bratislava,
8         (
9             SELECT ST_UNION(way)
10            FROM planet_osm_polygon
11            WHERE admin_level = '4'
12            GROUP BY admin_level
13        ) slovensko
14 )
15 INSERT INTO planet_osm_polygon(name, way)
16 VALUES(
17     'Okolie_Bratislavy',
18     (SELECT ST_DIFFERENCE(ST_INTERSECTION(ST_BUFFER(bratislava, 20000), slovensko), bratislava) FROM subset)
19 );
20
21
22
23 SELECT name, way
24 FROM planet_osm_polygon
25 WHERE admin_level = '4' OR name = 'Okolie_Bratislavy'
```



# Slovenská Technická Univerzita v Bratislave

## Fakulta Informatiky a Informačných Technológií



```
1 SELECT
2     name,
3     ST_AREA(ST_TRANSFORM(way, 4326), true) / 1000000 km_squared
4 FROM planet_osm_polygon
5 WHERE name = 'Okolie_Bratislavy';
```

Data output Messages Geometry Viewer × Notifications

	name text	km_squared double precision
1	Okolie_Bratislavy	885.875825132127