

## **BAZY DANYCH PRZESTRZENNYCH – ĆWICZENIA 6; POSTGIS RASTER**

Kroki postępowania:

1. Utworzenie nowej bazy danych w pgAdmin.
2. Utworzenie schematów,
3. Dodanie rozszerzeń: *postgis i postgis\_raster*.
4. Załadowanie kopii bazy danych przy użyciu *Restore*,
5. Załadowanie danych według instrukcji w pliku:

### **Przykład 1 – ładowanie rastru przy użyciu pliku .sql**

(dane zostały załadowane do wiersza poleceń bezpośrednio w folderu w którym znajduje się raster2pgsql.exe)

```
raster2pgsql.exe -s 3763 -N -32767 -t 100x100 -l -C -M -d "D:\AGH\semestr V\BDP roboczy\Dane -  
cwiczenia 6\rasters\srtm_1arc_v3.tif" rasters.dem > "D:\AGH\semestr V\BDP roboczy\Dane -  
cwiczenia 6\rasters\dem.sql"
```

### **Przykład 2 – ładowanie rastru bezpośrednio do bazy**

```
raster2pgsql.exe -s 3763 -N -32767 -t 100x100 -l -C -M -d "D:\AGH\semestr V\BDP roboczy\Dane -  
cwiczenia 6\rasters\srtm_1arc_v3.tif" rasters.dem | psql -d zajecia6 -h localhost -U postgres -p 5432
```

### **Przykład 3 – załadowanie danych landsat 8 o wielkości kafelka 128x128 bezpośrednio do bazy danych.**

```
raster2pgsql.exe -s 3763 -N -32767 -t 128x128 -l -C -M -d "D:\AGH\semestr V\BDP roboczy\Dane -  
cwiczenia 6\rasters\Landsat8_L1TP_RGBN.TIF" rasters.landsat8 | psql -d zajecia6 -h localhost -U  
postgres -p 5432
```

Kolejne etapy wykonano według poleceń opisanych w pliku pdf.

6. Tworzenie rastrów z istniejących rastrów i interakcja z wektorami.

- **Przykład 1- ST\_Intersects.**

Przecięcie rastra z wektorem. W tym przykładzie zostało pokazane jak wyodrębnić kafelki nakładające się na geometrię.

```
CREATE TABLE jalocho.intersects AS SELECT  
    a.rast, b.municipality FROM rasters.dem AS a,  
    vectors.porto_parishes AS b
```

```
WHERE ST_Intersects(a.rast, b.geom) AND b.municipality
ilike 'porto';
```

W przypadku tworzenia tabel zawierających dane rastrowe sugeruje się wykonanie poniższych kroków:

### 1. dodanie serial primary key:

```
alter table jalocha.intersects
add column rid SERIAL PRIMARY KEY;
```

### 2. utworzenie indeksu przestrzennego:

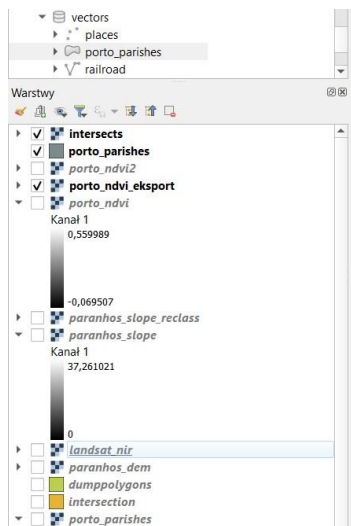
```
CREATE INDEX idx_intersects_rast_gist ON jalocha.intersects
USING gist (ST_ConvexHull(rast));
```

### 3. dodanie raster constraints:

```
(-- schema::name table_name::name raster_column::name)
```

```
SELECT
AddRasterConstraints('jalocha'::name, 'intersects'::name, 'rast'::name
);
```

Wizualizacja w QGIS:

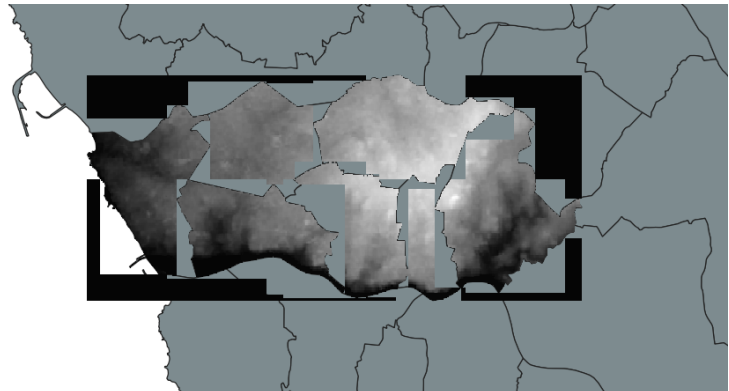
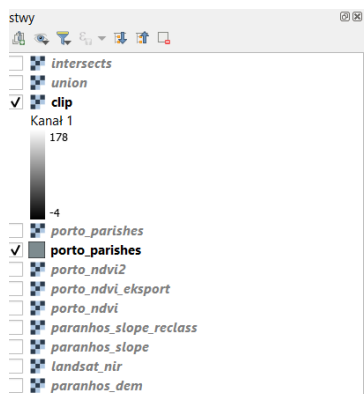


- **Przykład 2 - ST\_Clip**

Obcinanie rastra na podstawie wektora.

```
CREATE TABLE jalochoa.clip AS
SELECT ST_Clip(a.rast, b.geom, true), b.municipality
FROM rasters.dem AS a, vectors.porto_parishes AS b
WHERE ST_Intersects(a.rast, b.geom) AND b.municipality like 'PORTO';
```

Wizualizacja w QGIS:

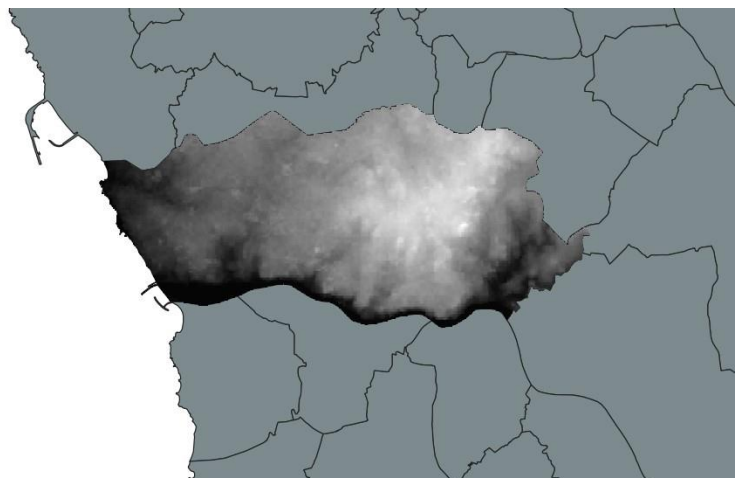
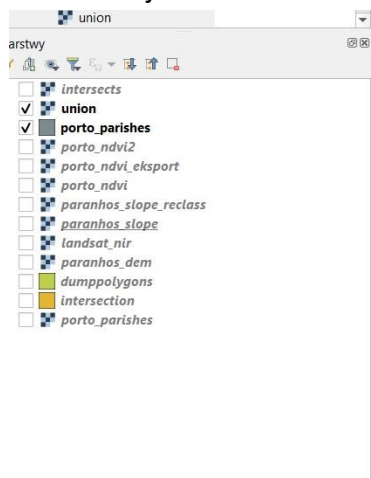


- **Przykład 3 - ST\_Union**

Połączenie wielu kafelków w jeden raster.

```
CREATE TABLE jalochoa.union AS
SELECT ST_Union(ST_Clip(a.rast, b.geom, true))
FROM rasters.dem AS a, vectors.porto_parishes AS b
WHERE b.municipality ilike 'porto' and ST_Intersects(b.geom,a.rast);
```

Wizualizacja:



## 7. Tworzenie rastrow z wektorów (rastrowanie).

### • Przykład 1 - ST\_AsRaster

Przykład pokazuje użycie funkcji ST\_AsRaster w celu rastrowania tabeli z parafiami o takiej samej charakterystyce przestrzennej tj.: wielkość piksela, zakresy itp.

```
CREATE TABLE jalocho.porto_parishes AS
WITH r AS (
SELECT rast FROM rasters.dem
LIMIT 1
)
SELECT ST_AsRaster(a.geom,r.rast,'8BUI',a.id,-32767) AS rast
FROM vectors.porto_parishes AS a, r
WHERE a.municipality ilike 'porto';
```

Wizualizacja:

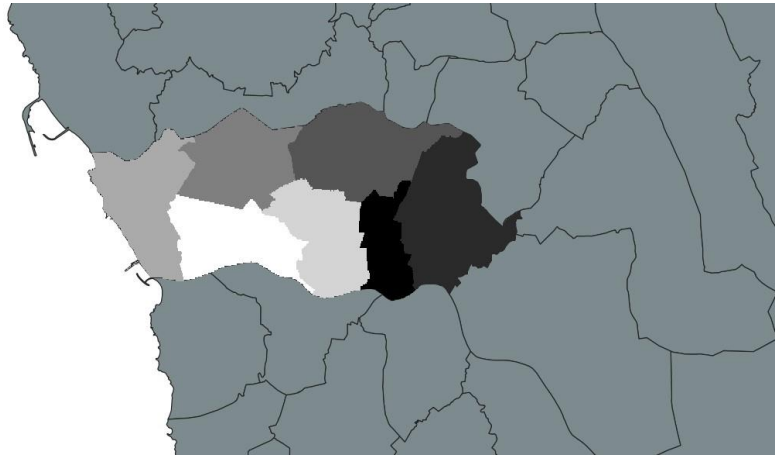
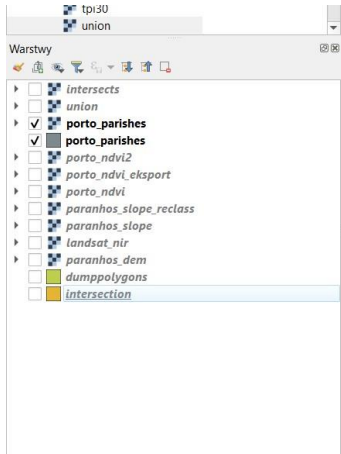


### • Przykład 2 - ST\_Union

Drugi przykład łączy rekordy z poprzedniego przykładu przy użyciu funkcji ST\_UNION w pojedynczy raster.

```
DROP TABLE jalocho.porto_parishes; --> drop table porto_parishes
first
CREATE TABLE jalocho.porto_parishes AS
WITH r AS (
SELECT rast FROM rasters.dem
LIMIT 1
)
SELECT st_union(ST_AsRaster(a.geom,r.rast,'8BUI',a.id,-32767)) AS
rast
FROM vectors.porto_parishes AS a, r
WHERE a.municipality ilike 'porto';
```

## Wizualizacja:

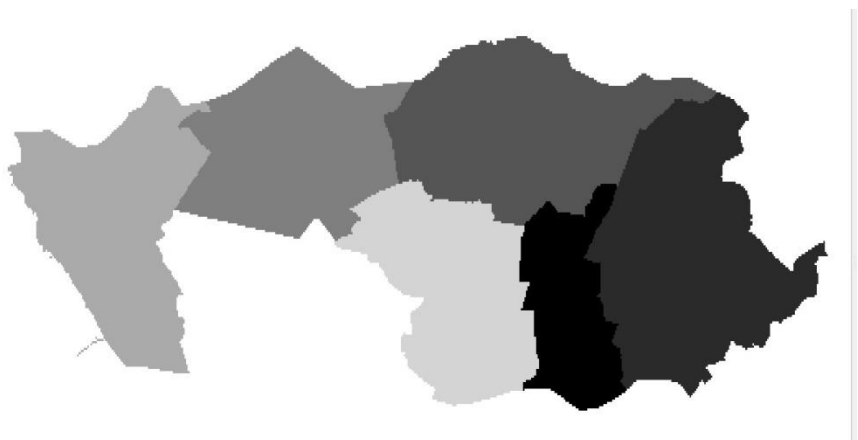
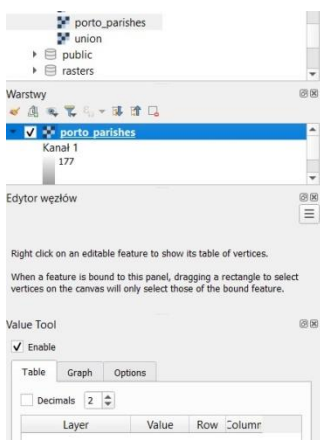


### • Przykład 3 - ST\_Tile

Po uzyskaniu pojedynczego rastra można generować kafelki za pomocą funkcji ST\_Tile.

```
DROP TABLE jalocho.porto_parishes; --> drop table porto_parishes
first
CREATE TABLE jalocho.porto_parishes AS
WITH r AS (
SELECT rast FROM rasters.dem
LIMIT 1 )
SELECT st_tile(st_union(ST_AsRaster(a.geom,r.rast,'8BUI',a.id,-
32767)),128,128,true,-32767) AS rast
FROM vectors.porto_parishes AS a, r
WHERE a.municipality ilike 'porto';
```

## Wizualizacja:



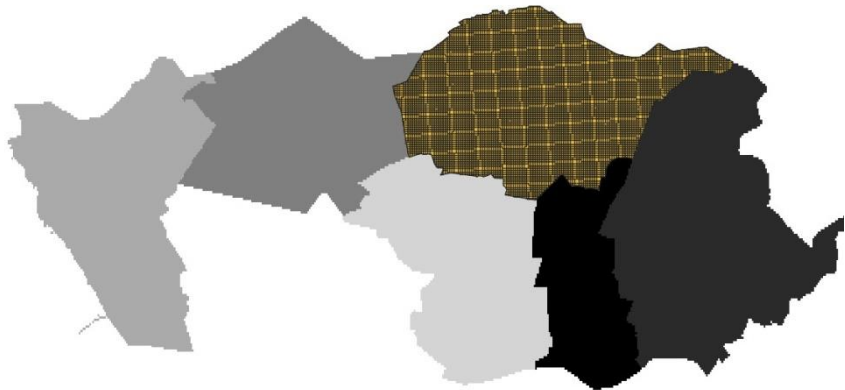
## 8. Konwertowanie rastrow na wektory (wektoryzowanie).

### • Przykład 1 - ST\_Intersection

Funkcja `St_Intersection` jest podobna do `ST_Clip`. `ST_Clip` zwraca raster, a `ST_Intersection` zwraca zestaw par wartości geometria-piksel, ponieważ ta funkcja przekształca raster w wektor przed rzeczywistym „klipem”. Zazwyczaj `ST_Intersection` jest wolniejsze od `ST_Clip` więc zasadnym jest przeprowadzenie operacji `ST_Clip` na rastrze przed wykonaniem funkcji `ST_Intersection`.

```
CREATE TABLE jalocha.intersection AS
SELECT
a.rid, (ST_Intersection(b.geom,a.rast)).geom, (ST_Intersection(b.geom,
a.rast)).val
FROM rasters.landsat8 AS a, vectors.porto_parishes AS b
WHERE b.parish ilike 'paranhos' and ST_Intersects(b.geom,a.rast);
```

Wizualizacja:

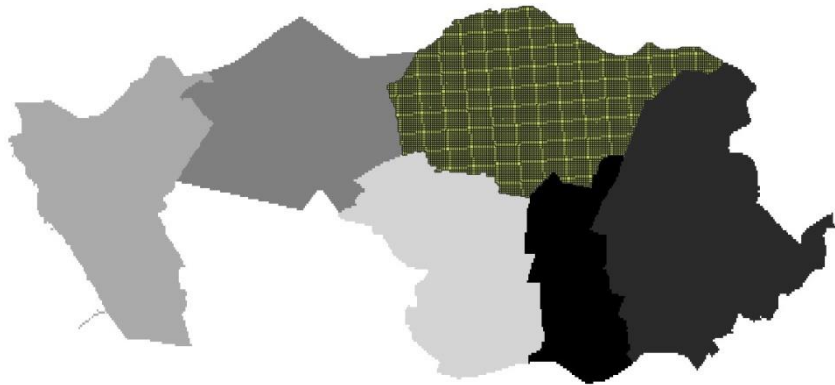
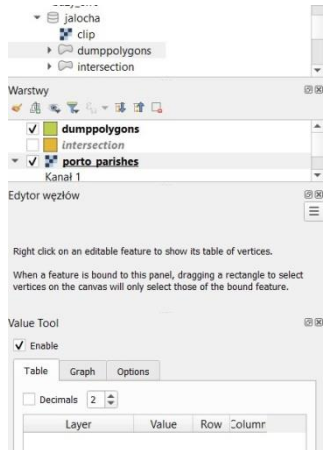


### • Przykład 2 - ST\_DumpAsPolygons

`ST_DumpAsPolygons` konwertuje rastry w wektory (poligony).

```
CREATE TABLE jalocha.dumppolygons AS
SELECT
a.rid, (ST_DumpAsPolygons(ST_Clip(a.rast,b.geom))).geom, (ST_DumpAsPol
ygons(ST_Clip(a.rast,b.geom))).val
FROM rasters.landsat8 AS a, vectors.porto_parishes AS b
WHERE b.parish ilike 'paranhos' and ST_Intersects(b.geom,a.rast);
```

## Wizualizacja:



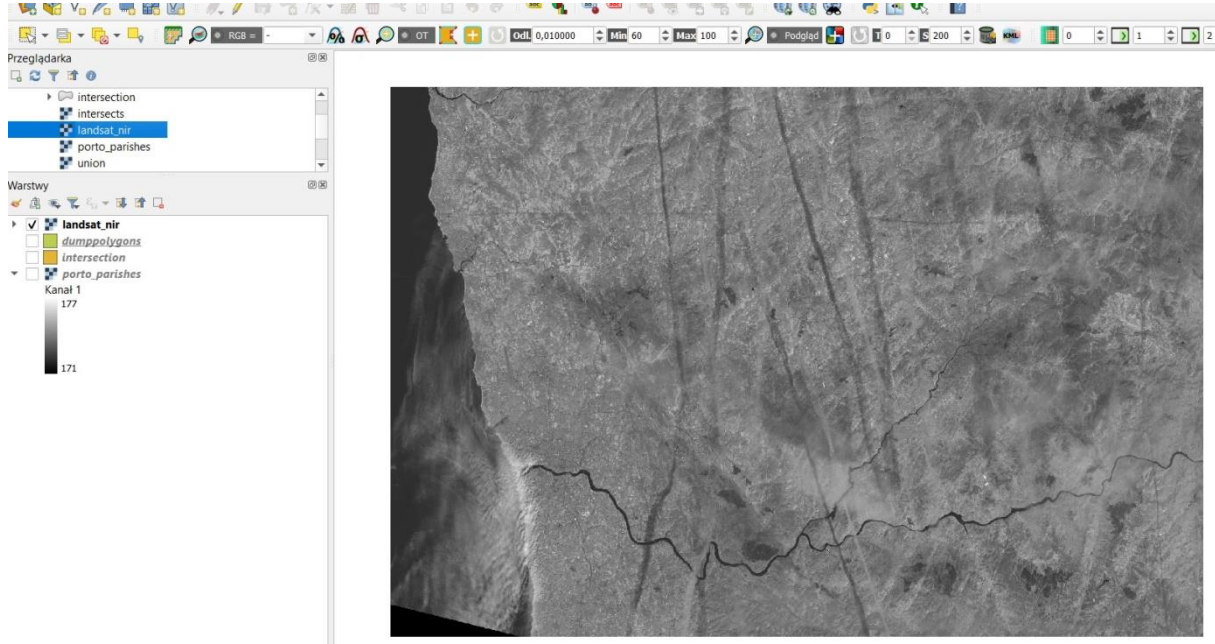
## 9. Analiza rastrów.

### • Przykład 1 - ST\_Band

Funkcja ST\_Band służy do wyodrębniania pasm z rastra.

```
CREATE TABLE jalocho.landsat_nir AS  
SELECT rid, ST_Band(rast,4) AS rast  
FROM rasters.landsat8;
```

## Wizualizacja:



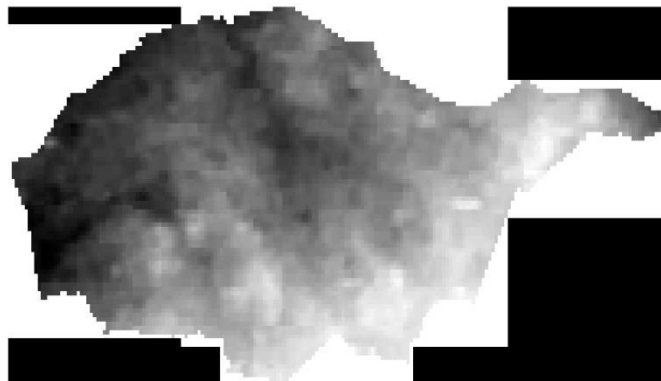
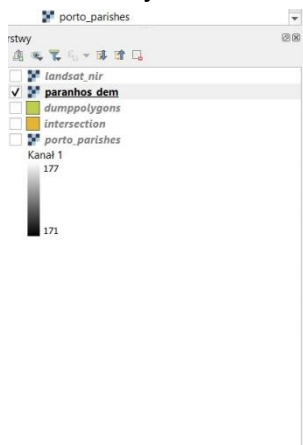


- **Przykład 2 - ST\_Clip**

ST\_Clip może być użyty do wycięcia rastra z innego rastra. Poniższy przykład wycina jedną parafię z tabeli vectors.porto\_parishes. Wynik będzie potrzebny do wykonania kolejnych przykładów.

```
CREATE TABLE jalocho.paranhos_dem AS
SELECT a.rid,ST_Clip(a.rast, b.geom,true) as rast
FROM rasters.dem AS a, vectors.porto_parishes AS b
WHERE b.parish ilike 'paranhos' and ST_Intersects(b.geom,a.rast);
```

Wizualizacja:

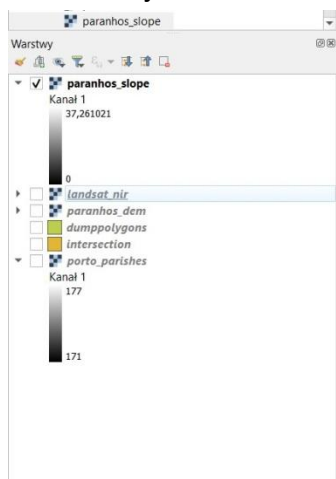


- **Przykład 3 - ST\_Slope**

Poniższy przykład użycia funkcji ST\_Slope wygeneruje nachylenie przy użyciu poprzednio wygenerowanej tabeli (wzniesienie).

```
CREATE TABLE jalocho.paranhos_slope AS
SELECT a.rid,ST_Slope(a.rast,1,'32BF','PERCENTAGE') as rast
FROM jalocho.paranhos_dem AS a;
```

Wizualizacja:





- **Przykład 4 - ST\_Reclass**

Aby zreklasifikować raster należy użyć funkcji ST\_Reclass.

```
CREATE TABLE jalocha.paranhos_slope_reclass AS
SELECT a.rid, ST_Reclass(a.rast, 1, ']'0-15]:1, (15-30]:2, (30-
9999:3', '32BF', 0)
FROM jalocha.paranhos_slope AS a;
```

Wizualizacja:



- **Przykład 5 - ST\_SummaryStats**

Aby obliczyć statystyki rastra można użyć funkcji ST\_SummaryStats. Poniższy przykład wygeneruje statystyki dla kafelka.

```
SELECT st_summarystats(a.rast) AS stats
FROM jalocha.paranhos_dem AS a;
```

- **Przykład 6 - ST\_SummaryStats oraz Union**

Przy użyciu UNION można wygenerować jedną statystykę wybranego rastra

```
SELECT st_summarystats(ST_Union(a.rast))
FROM jalocha.paranhos_dem AS a;
```

- **Przykład 7 - ST\_SummaryStats z lepszą kontrolą złożonego typu danych**

```
WITH t AS (
    SELECT st_summarystats(ST_Union(a.rast)) AS stats
    FROM jalocha.paranhos_dem AS a
)
SELECT (stats).min, (stats).max, (stats).mean FROM t;
```

- **Przykład 8 - ST\_SummaryStats w połączeniu z GROUP BY**

Aby wyświetlić statystykę dla każdego poligonu "parish" można użyć polecenia GROUP BY

```
WITH t AS (
    SELECT b.parish AS parish,
    st_summarystats(ST_Union(ST_Clip(a.rast,b.geom,true))) AS stats
    FROM rasters.dem AS a, vectors.porto_parishes AS b
    WHERE b.municipality ilike 'porto' and
    ST_Intersects(b.geom,a.rast) group by b.parish
)
SELECT parish,(stats).min,(stats).max,(stats).mean FROM t;
```

- **Przykład 9 - ST\_Value**

Funkcja ST\_Value pozwala wyodrębnić wartość piksela z punktu lub zestawu punktów.

Poniższy przykład wyodrębnia punkty znajdujące się w tabeli vectors.places.

Ponieważ geometria punktów jest wielopunktowa, a funkcja ST\_Value wymaga geometrii jednopunktowej, należy przekonwertować geometrię wielopunktową na geometrię jednopunktową za pomocą funkcji (ST\_Dump(b.geom)).geom.

```
SELECT b.name,st_value(a.rast,(ST_Dump(b.geom)).geom)
FROM
rasters.dem a, vectors.places AS b
WHERE ST_Intersects(a.rast,b.geom)
ORDER BY b.name;
```

- **Przykład 10 - ST\_TPI**

TPI porównuje wysokość każdej komórki w DEM ze średnią wysokością określonego sąsiedztwa wokół tej komórki. Wartości dodatnie reprezentują lokalizacje, które są wyższe niż średnia ich otoczenia, zgodnie z definicją sąsiedztwa (grzbietów). Wartości ujemne reprezentują lokalizacje, które są niższe niż ich otoczenie (doliny).

Funkcja ST\_Value pozwala na utworzenie mapy TPI z DEM wysokości.

```
CREATE TABLE jalocho.tpi30 AS
SELECT ST_TPI(a.rast,1) as rast
FROM rasters.dem a;
```

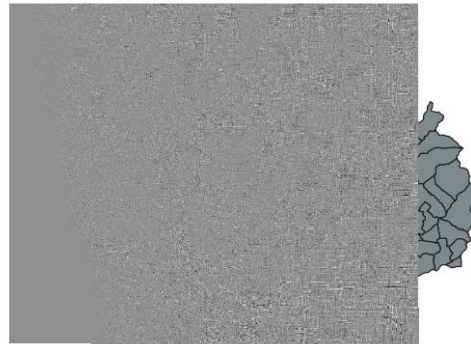
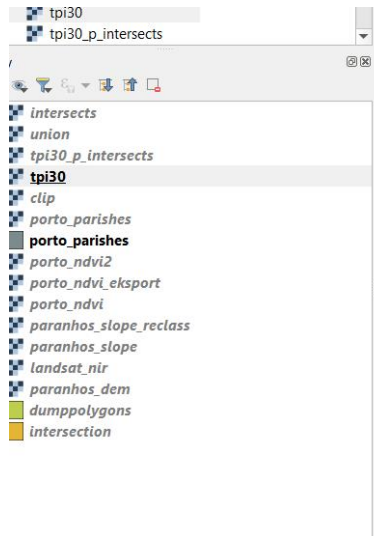
--Poniższa kwerenda utworzy indeks przestrzenny:

```
CREATE INDEX idx_tpi30_rast_gist ON jalocho.tpi30
USING gist (ST_ConvexHull(rast));
```

--Dodanie constraintów:

```
SELECT
AddRasterConstraints('jalocho'::name,'tpi30'::name,'rast'::name);
```

## Wizualizacja:



## Problem do samodzielnego rozwiązania

Przetwarzanie poprzedniego zapytania może potrwać dłużej niż minutę, a niektóre zapytania mogą potrwać zbyt długo. W celu skrócenia czasu przetwarzania czasami można ograniczyć obszar zainteresowania i obliczyć mniejszy region. Dostosuj zapytanie z przykładu 10, aby przetwarzać tylko gminę Porto. Musisz użyć ST\_Intersects, sprawdź Przykład 1 - ST\_Intersects w celach informacyjnych. Porównaj różne czasy przetwarzania. Na koniec sprawdź wynik w QGIS.

```
CREATE TABLE jalocha.tpi30_p_intersects AS
SELECT ST_TPI(a.rast,1) AS rast
FROM rasters.dem AS a, vectors.porto_parishes AS b
WHERE ST_Intersects(a.rast, b.geom) AND b.municipality ILIKE
'porto';
```

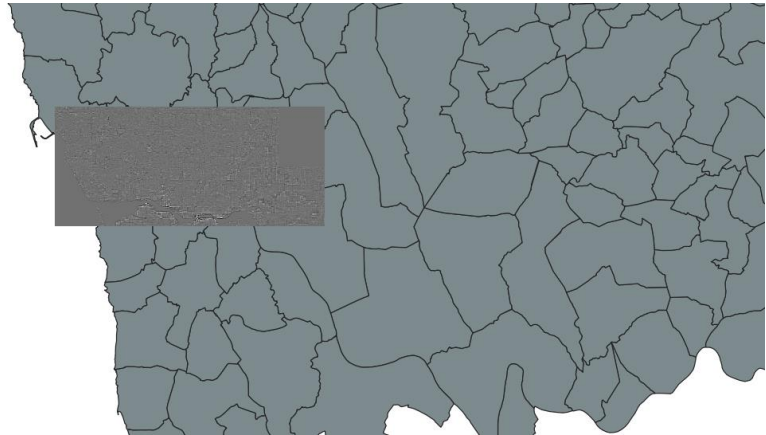
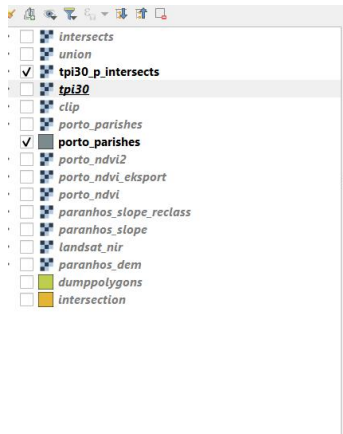
Indeks przestrzenny:

```
CREATE INDEX idx_tpi30_rast_gist ON jalocha.tpi30_p_intersects
USING gist (ST_ConvexHull(rast));
```

Dodanie constraintów:

```
SELECT
AddRasterConstraints('jalocha'::name, 'tpi30_p_intersects'::name, 'rast'::name);
```

## Wizualizacja:



## 10. Algebra map.

### • Przykład 1 - Wyrażenie Algebry Map.

--Wzór na NDVI:

--NDVI=(NIR-Red)/(NIR+Red)

```
CREATE TABLE jalocho.porto_ndvi AS
WITH r AS (
    SELECT a.rid, ST_Clip(a.rast, b.geom, true) AS rast
    FROM rasters.landsat8 AS a, vectors.porto_parishes AS b
    WHERE b.municipality ilike 'porto' and
    ST_Intersects(b.geom, a.rast)
)
SELECT
    r.rid, ST_MapAlgebra(
        r.rast, 1,
        r.rast, 4,
        '([rast2.val] - [rast1.val]) /
        ([rast2.val] + [rast1.val])::float', '32BF'
    ) AS rast
FROM r;
```

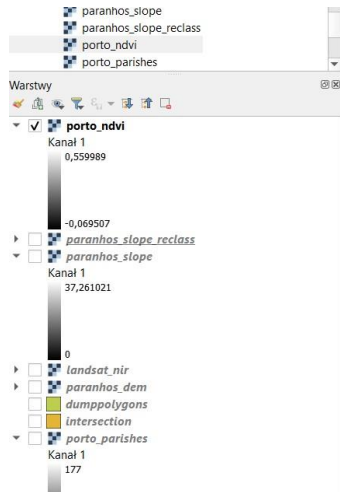
--Poniższe zapytanie utworzy indeks przestrzenny na wcześniej stworzonej tabeli:

```
CREATE INDEX idx_porto_ndvi_rast_gist ON jalocho.porto_ndvi
USING gist (ST_ConvexHull(rast));
```

--Dodanie constraintów:

```
SELECT
AddRasterConstraints('jalocho'::name, 'porto_ndvi'::name, 'rast'::name
);
```

## Wizualizacja:



### • Przykład 2 – Funkcja zwrotna

W pierwszym kroku należy utworzyć funkcję, które będzie wywołana później:

```
CREATE OR REPLACE FUNCTION jalocho.ndvi(
    value double precision [] [] [],
    pos integer [],
    VARIADIC userargs text []
)
RETURNS double precision AS
$$
BEGIN
    --RAISE NOTICE 'Pixel Value: %', value [1][1][1];-->For debug
    purposes
    RETURN (value [2][1][1] - value [1][1][1])/(value
    [2][1][1]+value[1][1][1]); --> NDVI calculation!
END;
$$
LANGUAGE 'plpgsql' IMMUTABLE COST 1000;
```

W kwerendzie algebry map należy można wywołać zdefiniowaną wcześniej funkcję:

```
CREATE TABLE jalocho.porto_ndvi2 AS
WITH r AS (
    SELECT a.rid,ST_Clip(a.rast, b.geom,true) AS rast
    FROM rasters.landsat8 AS a, vectors.porto_parishes AS b
    WHERE b.municipality ilike 'porto' and
    ST_Intersects(b.geom,a.rast)
)
SELECT
    r.rid,ST_MapAlgebra(
```

*FROM*  $r$ ;

*Dodanie indeksu przestrzennego:*

```
USING gist (ST_ConvexHull(rast));
```

*Dodanie constraintów:*

e) ;

Wizualizacja:



## 11. Eksport danych.

- **Przykład 1 - ST\_AsTiff**

```
FROM jalocha.porto ndvi;
```

- **Przykład 2 - ST\_AsGDALRaster**

```
'PREDICTOR=2', 'PZLEVEL=9']) FROM jalocha.porto ndvi;
```

- **Przykład 3 - Zapisywanie danych na dysku za pomocą dużego obiektu (large object, lo)**

```
CREATE TABLE tmp_out AS
SELECT lo_from_bytea(0,ST_AsGDALRaster(ST_Union(rast), 'GTiff',
ARRAY['COMPRESS=DEFLATE', 'PREDICTOR=2', 'PZLEVEL=9'])
) AS loid
FROM jalocho.porto_ndvi;

-----

SELECT lo_export(loid, 'F:\KAMSSSSSS\bazy_obiekt') --> Save the file
in a place where the user postgres have access. In windows a flash
drive usually works fine.

FROM tmp_out;

-----

SELECT lo_unlink(loid)
FROM tmp_out; --> Delete the large object.

DROP TABLE tmp_out;
```

- **Przykład 4 - Użycie Gdal**

```
gdal_translate -co COMPRESS=DEFLATE -co PREDICTOR=2 -co ZLEVEL=9
PG:"host=localhost port=5432 dbname=zajecia6 user=postgres
password=***** schema=jalocho table=porto_ndvi mode=2"
porto_ndvi.tiff
```

- **Publikowanie danych za pomocą MapServer**

```
MAP
  NAME 'map'
  SIZE 800 650
  STATUS ON
  EXTENT -58968 145487 30916 206234
  UNITS METERS
  WEB
  METADATA
    'wms_title' 'Terrain wms'
    'wms_srs' 'EPSG:3763 EPSG:4326 EPSG:3857'
    'wms_enable_request' '*'
    'wms_onlineresource'
    'http://54.37.13.53/mapservices/srtm'
  END
```



```
END
    PROJECTION
'init=epsg:3763'
END
LAYER
    NAME srtm
    TYPE raster
    STATUS OFF
    DATA "PG:host=localhost port=5432 dbname='zajecia6' user='postgres'
password='*****' schema='rasters' table='dem' mode='2'" PROCESSING
"SCALE=AUTO"
    PROCESSING "NODATA=-32767"
    OFFSITE 0 0 0
    METADATA
'wms_title' 'srtm'
        END
    END
END
```