Ładowanie danych rastrowych

--dem sql

raster2pgsql.exe -s 3763 -N -32767 -t 100x100 -I -C -M -d E:\semestr5\Bazy_danych_przestrzennych\cw6\rasters\srtm_1arc_v3.tif rasters.dem > E:\semestr5\Bazy_danych_przestrzennych\cw6\dem.sql

--dem to postgis

raster2pgsql.exe -s 3763 -N -32767 -t 100x100 -I -C -M -d E:\semestr5\Bazy_danych_przestrzennych\cw6\rasters\srtm_1arc_v3.tif rasters.dem | psql -d rastry -h localhost -U postgres -p 5432

--landsat to postgis

raster2pgsql.exe -s 3763 -N -32767 -t 128x128 -I -C -M -d E:\semestr5\Bazy_danych_przestrzennych\cw6\rasters\Landsat8_L1TP_RGBN.TIF rasters.landsat8 | psql -d rastry -h localhost -U postgres -p 5432

Tworzenie rastrów z istniejących rastrów i interakcja z wektorami

--Przykład 1

CREATE TABLE schema_kozlowski.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';

- 1. dodanie serial primary key:

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

--2. utworzenie indeksu przestrzennego:
CREATE INDEX idx_intersects_rast_gist ON schema_kozlowski.intersects
USING gist (ST_ConvexHull(rast));

--3. dodanie raster constraints:
-- schema::name table_name::name raster_column::name
SELECT AddRasterConstraints('schema_kozlowski'::name,
'intersects'::name,'rast'::name);
```

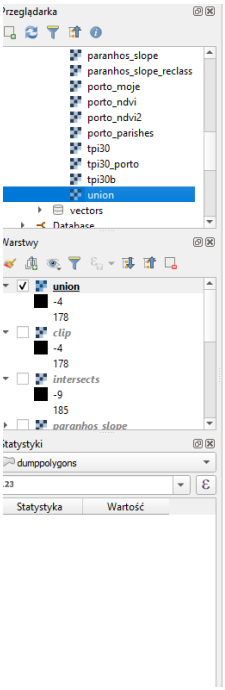
--Przykład 2 - ST_Clip

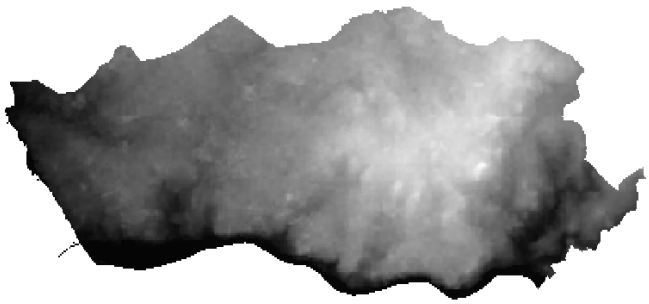
```
--Obcinanie rastra na podstawie wektora.
CREATE TABLE schema_kozlowski.clip AS
SELECT ST_Clip(a.rast, b.geom, true), b.municipality
FROM rasters.dem AS a, vectors.porto_parishes AS b
WHERE ST_Intersects(a.rast, b.geom) AND b.municipality like 'PORTO';
```

--Przykład 3 - ST_Union

--Połączenie wielu kafelków w jeden raster.

```
CREATE TABLE schema_kozlowski.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);
```





-- Tworzenie rastrów z wektorów (rastrowanie)

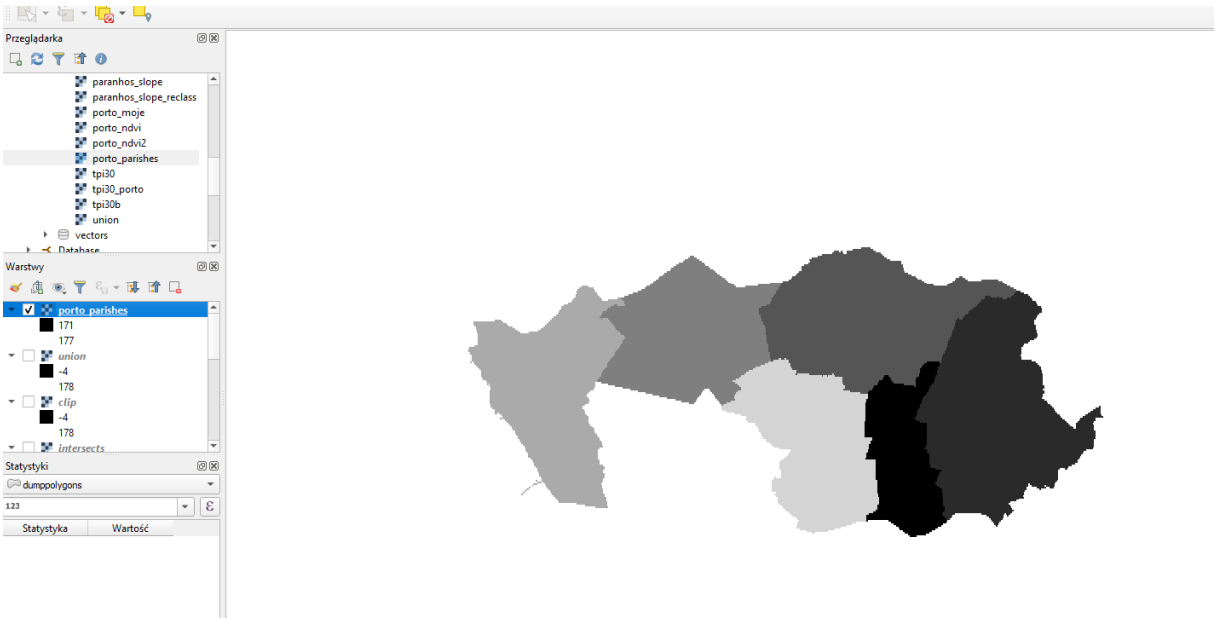
-- Przykład 1 - ST_AsRaster

-- Przykład pokazuje użycie funkcji ST_AsRaster w celu rastrowania tabeli z parafiami o takiej samej

-- charakterystyce przestrzennej tj.: wielkość piksela, zakresy itp.

```
CREATE TABLE schema_kozlowski.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';



-- Przykład 2 - ST_Union

-- Wynikowy raster z poprzedniego zadania to jedna parafia na rekord, na wiersz tabeli. Użyj QGIS lub

-- ArcGIS do wizualizacji wyników.

-- Drugi przykład łączy rekordy z poprzedniego przykładu przy użyciu funkcji ST_UNION w pojedynczy

-- raster.

DROP TABLE schema_kozlowski.porto_parishes; --> drop table porto_parishes first

CREATE TABLE schema_kozlowski.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';

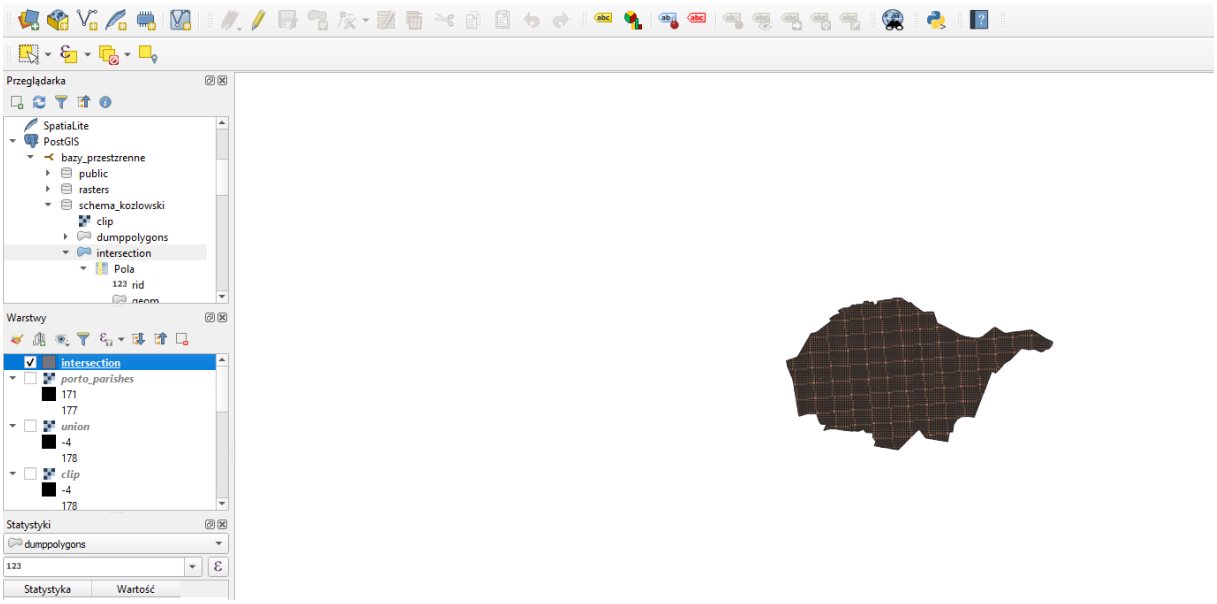
-- Przykład 3 - ST_Tile

-- Po uzyskaniu pojedynczego rastra można generować kafelki za pomocą funkcji ST_Tile.

```
DROP TABLE schema_kozlowski.porto_parishes; --> drop table porto_parishes first

CREATE TABLE schema_kozlowski.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';
```

Przykład 1 - ST_Intersection

```
create table schema_kozlowski.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);
```



-- Przykład 2 - ST_DumpAsPolygons

-- ST_DumpAsPolygons konwertuje rastry w wektory (poligony).

CREATE TABLE schema_kozlowski.dumppolygons AS

SELECT

a.rid,(ST_DumpAsPolygons(ST_Clip(a.rast,b.geom))).geom,(ST_DumpAsPolygons(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);

Analiza rastrów

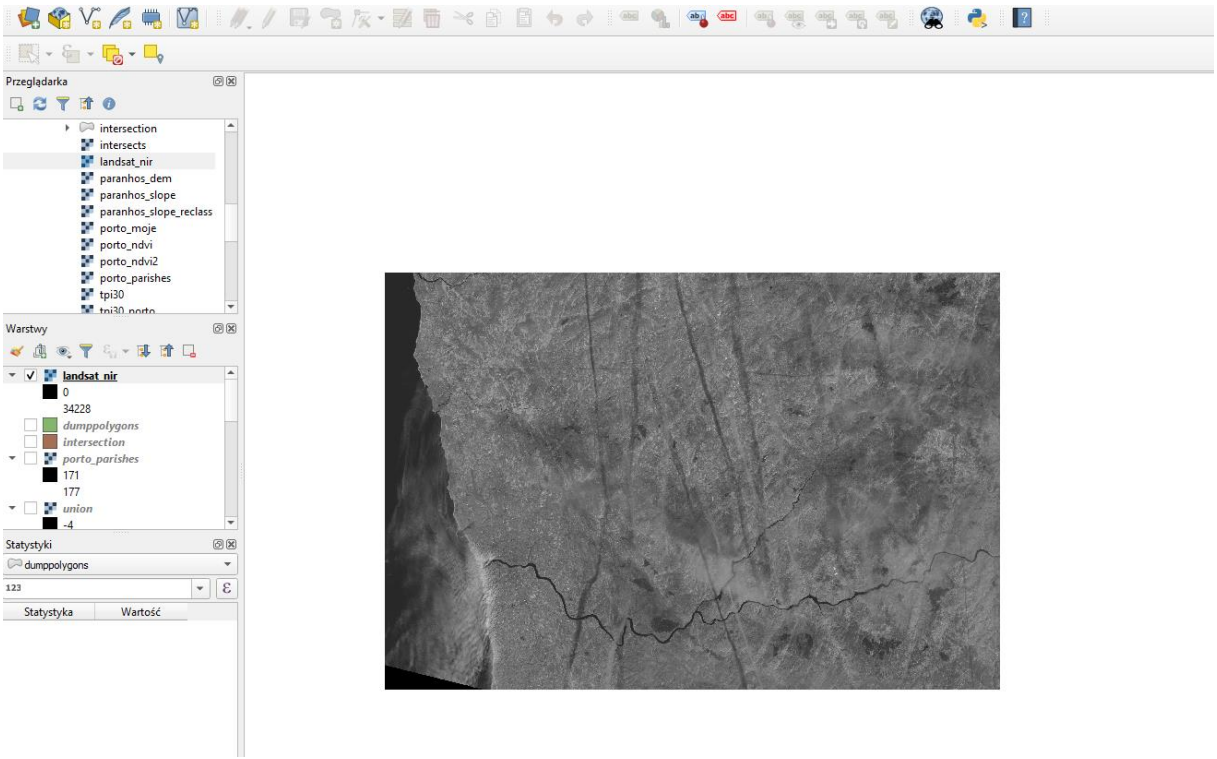
--Przykład 1 - ST_Band

--Funkcja ST_Band służy do wyodrębniania pasm z rastra

CREATE TABLE schema_kozlowski.landsat_nir AS

SELECT rid, ST_Band(rast,4) AS rast

FROM rasters.landsat8;



-- 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.

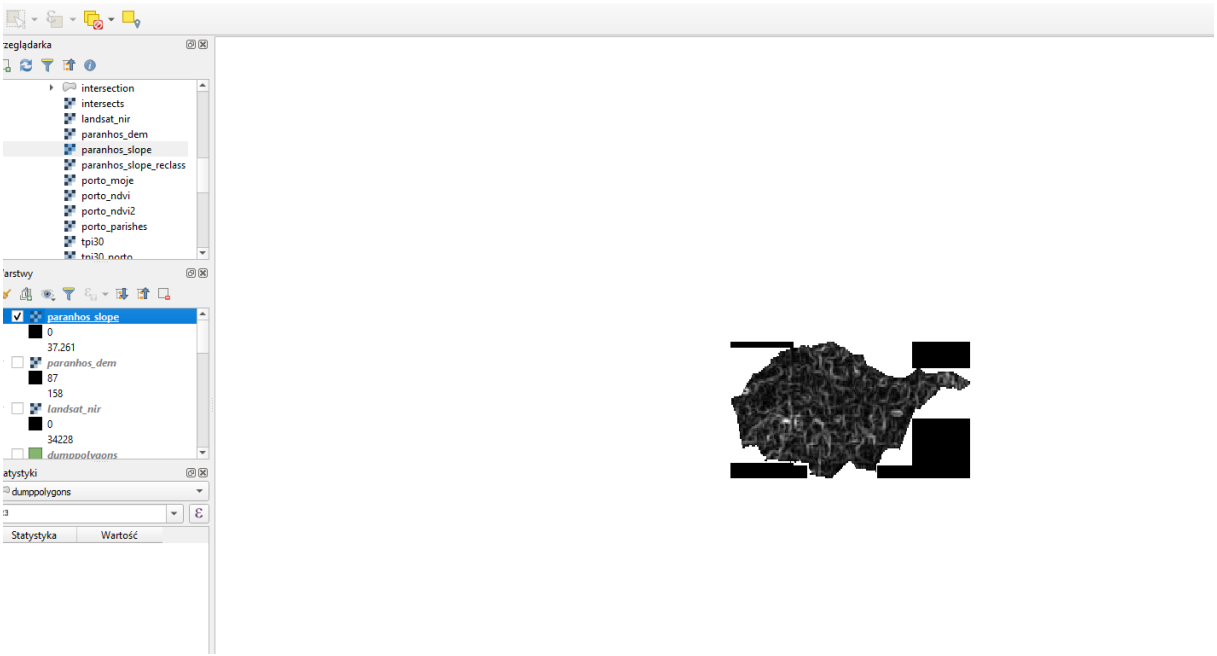
```
CREATE TABLE schema_kozlowski.paranhos_dem AS
SELECT a.rid,ST_Clip(a.rast, b.geom,true) as rast
FROM rasters.dem AS a, vectors.porto_parishes AS b
WHERE b.parish ilike 'paranhos' and ST_Intersects(b.geom,a.rast);
```

-- Przykład 3 - ST_Slope

-- Poniższy przykład użycia funkcji ST_Slope wygeneruje nachylenie przy użyciu poprzednio

-- wygenerowanej tabeli (wzniesienie).

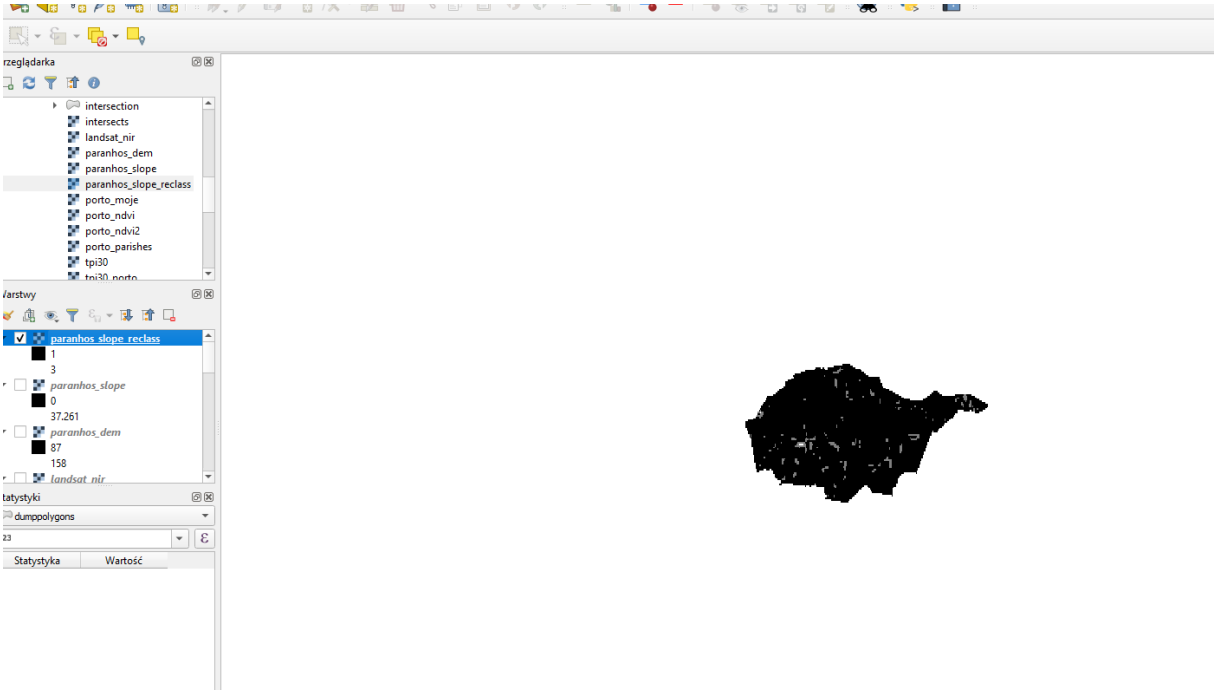
```
CREATE TABLE schema_kozlowski.paranhos_slope AS

SELECT a.rid,ST_Slope(a.rast,1,'32BF','PERCENTAGE') as rast

FROM schema_kozlowski.paranhos_dem AS a;
```



-- Przykład 4 - ST_Reclass

-- Aby zreklasyfikować raster należy użyć funkcji ST_Reclass.

```
CREATE TABLE schema_kozlowski.paranhos_slope_reclass AS

SELECT a.rid,ST_Reclass(a.rast,1,']0-15]:1, (15-30]:2, (30-9999:3', '32BF',0)

FROM schema_kozlowski.paranhos_slope AS a;
```



-- 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 schema_kozlowski.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 schema_kozlowski.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 schema_kozlowski.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

```
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
```

Topographic Position Index (TPI)

--Przykład 10 - ST_TPI

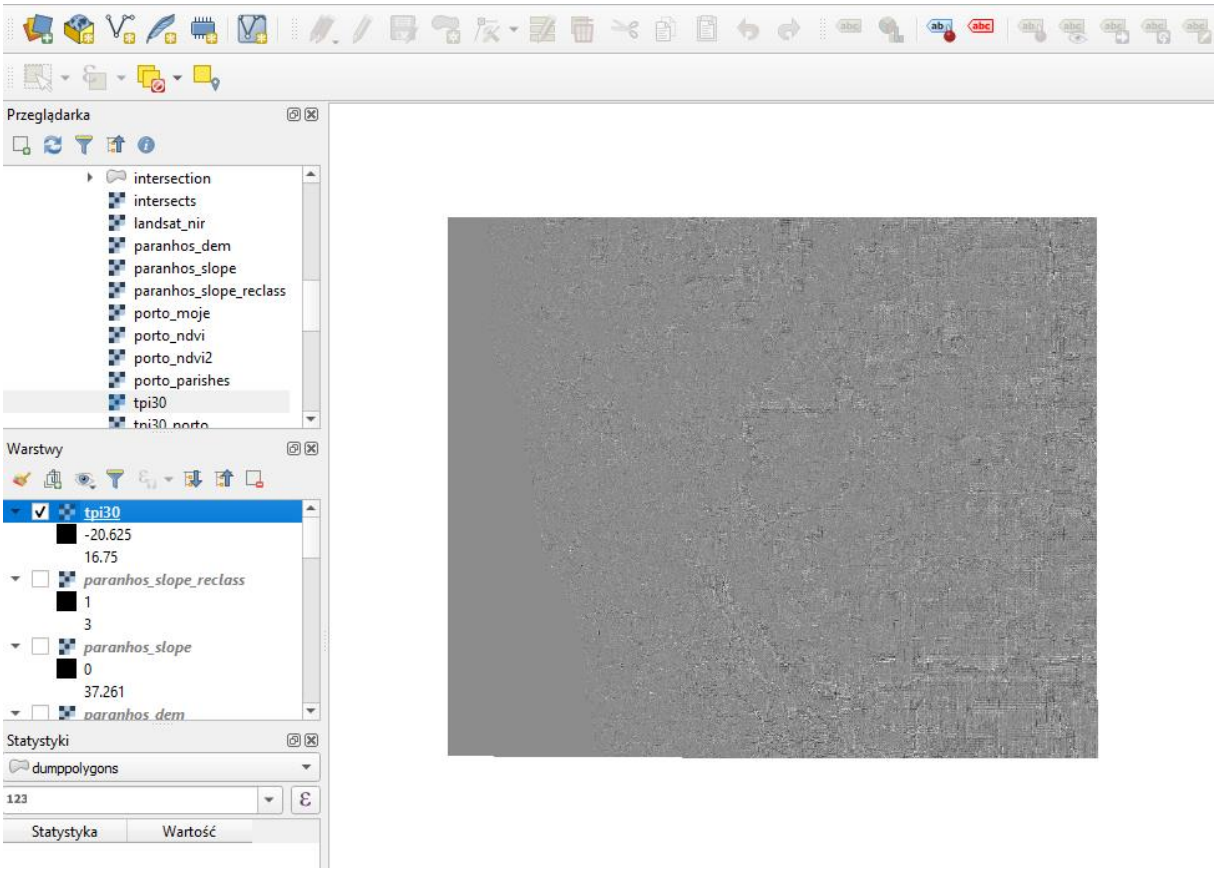
```
create table schema_kozlowski.tpi30 as
select ST_TPI(a.rast,1) as rast
from rasters.dem a;
-- Poniższa kwerenda utworzy indeks przestrzenny:
CREATE INDEX idx_tpi30_rast_gist ON schema_kozlowski.tpi30
```

USING gist (ST_ConvexHull(rast));

-- Dodanie constraintów:

SELECT AddRasterConstraints('schema_kozlowski'::name,

'tpi30'::name,'rast'::name);



Problem do samodzielnego rozwiązania

create table schema_kozlowski.tpi30b 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'

-- Dodanie indeksu przestrzennego:

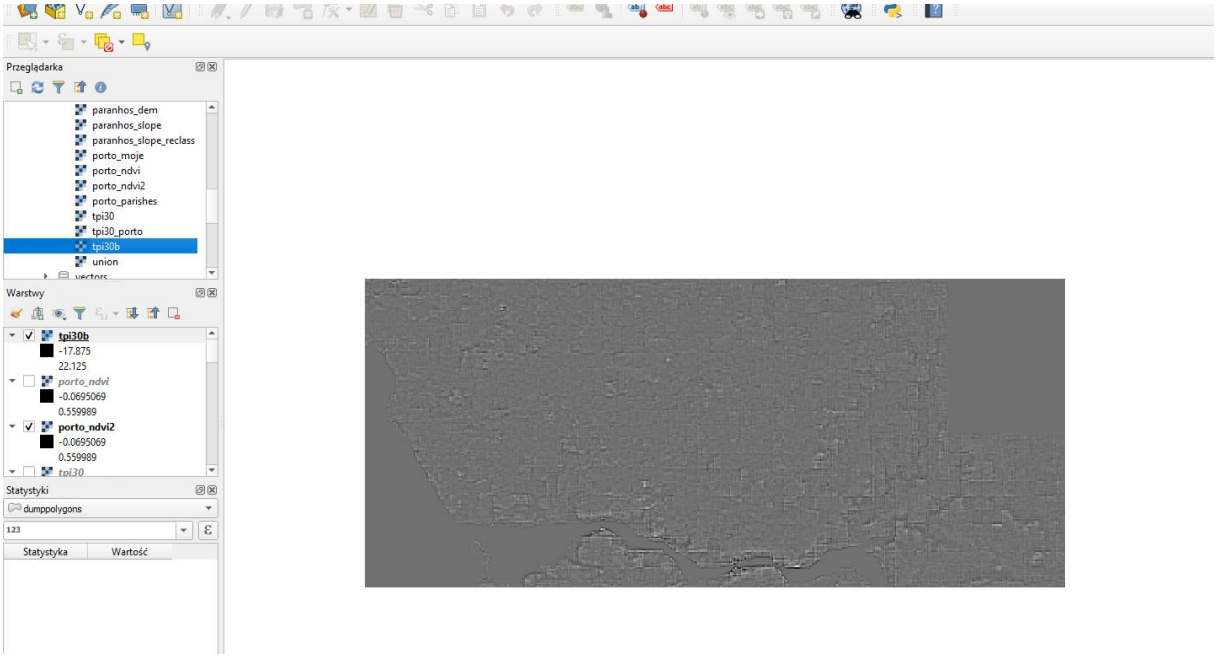
CREATE INDEX idx_tpi30b_rast_gist ON schema_kozlowski.tpi30b

USING gist (ST_ConvexHull(rast));

-- Dodanie constraintów:

SELECT AddRasterConstraints('schema_kozlowski'::name,

'tpi30b'::name,'rast'::name);



Algebra map

-- Przykład 1 - Wyrażenie Algebry Map

CREATE TABLE schema_kozlowski.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,

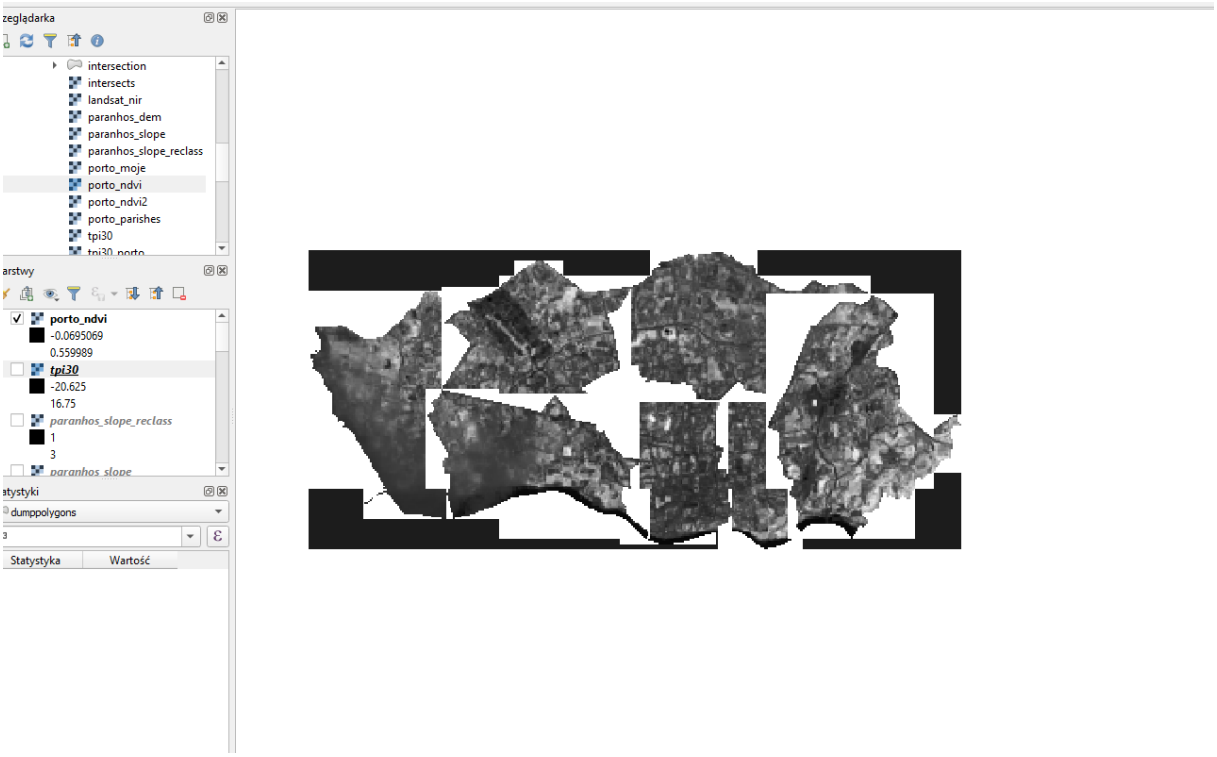
r.rast, 4,

'([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 schema_kozlowski.porto_ndvi

USING gist (ST_ConvexHull(rast));

-- Dodanie constraintów:

SELECT AddRasterConstraints('schema_kozlowski'::name,

'porto_ndvi'::name,'rast'::name);

-- Przykład 2 – Funkcja zwrotna

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

create or replace function schema_kozlowski.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 schema_kozlowski.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(
r.rast, ARRAY[1,4],
'schema_kozlowski.ndvi(double precision[],
integer[],text[])'::regprocedure, --> This is the function!
'32BF'::text
) AS rast
FROM r;
```

-- Dodanie indeksu przestrzennego:

CREATE INDEX idx_porto_ndvi2_rast_gist ON schema_kozlowski.porto_ndvi2

USING gist (ST_ConvexHull(rast));

-- Dodanie constraintów:

SELECT AddRasterConstraints('schema_kozlowski'::name,

'porto_ndvi2'::name,'rast'::name);

Eksport danych

-- Przykład 1 - ST_AsTiff

SELECT ST_AsTiff(ST_Union(rast))

FROM schema_kozlowski.porto_ndvi;

-- Przykład 2 - ST_AsGDALRaster

SELECT ST_AsGDALRaster(ST_Union(rast), 'GTiff', ARRAY['COMPRESS=DEFLATE',

'PREDICTOR=2', 'PZLEVEL=9'])

FROM schema_kozlowski.porto_ndvi;

-- Uwaga:

-- Funkcje ST_AsGDALRaster pozwalają nam zapisać raster w dowolnym formacie obsługiwanym przez

-- gdal. Aby wyświetlić listę formatów obsługiwanych przez bibliotekę uruchom:

SELECT ST_GDALDrivers();

-- 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 schema_kozlowski.porto_ndvi;

SELECT lo_export(loid, 'E:\semestr5\Bazy_danych_przestrzennych\cw6\myraster.tiff') --> Save the file in a place

-- where the user postgres have access. In windows a flash drive usualy works

-- fine.

FROM tmp_out;

SELECT lo_unlink(loid)

FROM tmp_out; --> Delete the large object.

-- Przykład 4 - Użycie Gdal

-- Gdal obsługuje rastry z PostGISa. Polecenie gdal_translate eksportuje raster do dowolnego formatu

-- obsługiwanego przez GDAL.

gdal_translate -co COMPRESS=DEFLATE -co PREDICTOR=2 -co ZLEVEL=9

PG:"host=localhost port=5432 dbname=postgis_raster user=postgres

password=postgis schema=schema_kozlowski table=porto_ndvi mode=2"

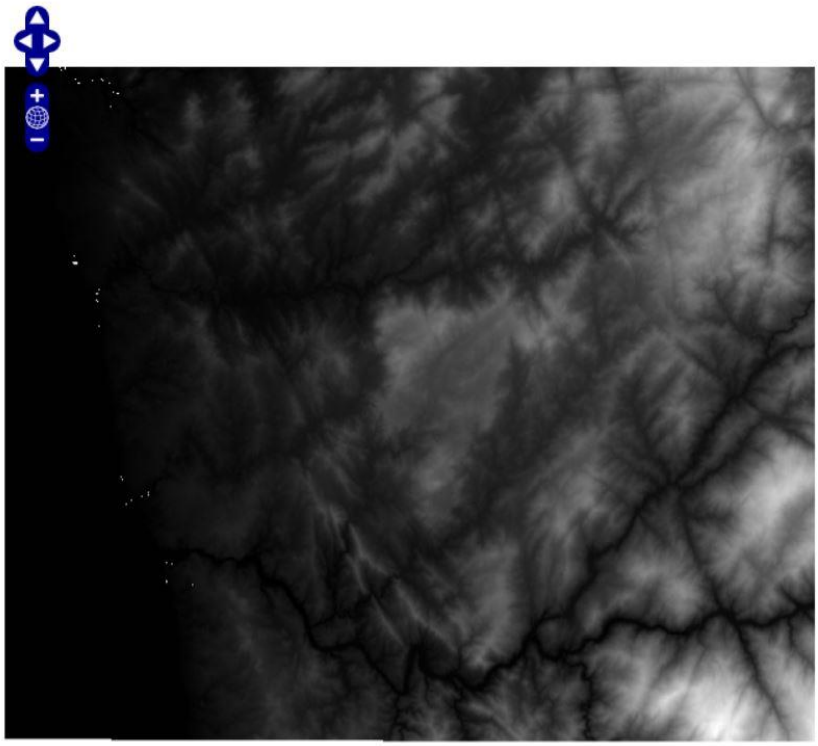
porto_ndvi.tiff

-- Publikowanie danych za pomocą MapServer

-- Przykład 1 - Mapfile

```
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='postgis_raster' user='sasig'
password='postgis' schema='rasters' table='dem' mode='2'"
                PROCESSING "SCALE=AUTO"
                PROCESSING "NODATA=-32767"
```

```
        OFFSITE 0 0 0
        METADATA
            'wms_title' 'srtm'
        END
    END
END
```



Publikowanie danych przy użyciu GeoServera


GeoServer: Podgląd warstw
OpenLayers map preview
localhost:8080/geoserver/pt.geomaster/wms
Scale = 1 : 68K
-35617.95652, 166923.38143
Click on the map to get feature info