Ćwiczenie 6 – PostGIS raster

Jakub Hempel, nr 405756

GEOINFORMATYKA, 3 rok

-- Zmiana nazwy schematu schema_name

ALTER SCHEMA schema_name RENAME TO hempel;

ZAŁADOWANIE DANYCH RASTROWYCH PRZY POMOCY raster2pgsql.exe (D:\PostgreSQL\14\bin)

-- Przykład 1. – Ładowanie rastru przy użyciu pliku .sql

raster2pgsql.exe -s 3763 -N -32767 -t 100x100 -I -C -M -d "C:\Users\kubah\Desktop\AGH zajecia\Bazy danych przestrzennych laborki\Cwiczenia_6\rasters\srtm_1arc_v3.tif" rasters.dem > „C:\Users\kubah\Desktop\AGH zajecia\Bazy danych przestrzennych laborki\Cwiczenia_6\rasters\dem.sql"

-- Przykład 2. – Ładowanie rastru bezpośrednio do bazy

raster2pgsql.exe -s 3763 -N -32767 -t 100x100 -I -C -M -d "C:\Users\kubah\Desktop\AGH zajecia\Bazy danych przestrzennych laborki\Cwiczenia_6\rasters\srtm_1arc_v3.tif" rasters.dem | psql -U postgres -d cwiczenia6 -h localhost -p 5432

```
D:\PostgreSQL\14\bin>raster2pgsql.exe -s 3763 -N -32767 -t 100x100 -I -C -M -d "C:\Users\kubah\D
esktop\AGH zajecia\Bazy danych przestrzennych laborki\Cwiczenia_6\rasters\srtm_1arc_v3.tif" rast
ers.dem | psql -U postgres -d cwiczenia6 -h localhost -p 5432
Password for user postgres: Processing 1/1: C:\Users\kubah\Desktop\AGH zajecia\Bazy danych przes
trzennych laborki\Cwiczenia_6\rasters\srtm_1arc_v3.tif

BEGIN
NOTICE:  table "dem" does not exist, skipping
DROP TABLE
CREATE TABLE
INSERT 0 1
INSERT 0 1
INSERT 0 1
```

```
INSERT 0 1
CREATE INDEX
ANALYZE
NOTICE:  Adding SRID constraint
NOTICE:  Adding scale-X constraint
NOTICE:  Adding scale-Y constraint
NOTICE:  Adding blocksize-X constraint
NOTICE:  Adding blocksize-Y constraint
NOTICE:  Adding alignment constraint
NOTICE:  Adding number of bands constraint
NOTICE:  Adding pixel type constraint
NOTICE:  Adding nodata value constraint
NOTICE:  Adding out-of-database constraint
NOTICE:  Adding maximum extent constraint
 addrasterconstraints
----------------------
 t
(1 row)

COMMIT
VACUUM
```

-- Przykład 3. – Załadowanie danych landsat 8 o wielkości kafelka 128x128 bezpośrednio do bazy danych

raster2pgsql.exe -s 3763 -N -32767 -t 128x128 -I -C -M -d “C:\Users\kubah\Desktop\AGH zajecia\Bazy danych przestrzennych laborki\Cwiczenia_6\rasters\Landsat8_L1TP_RGBN.TIF” rasters.landsat8 | psql -d cwiczenia6 -h localhost -U postgres -p 5432

Wynik taki sam jak na powyższym screenie.

TWORZENIE RASTRÓW Z ISTNIEJĄCYCH RASTRÓW Z INTERAKCJĄ Z WEKTORAMI

-- Przykład 1. - Przycięcie rastra z wektorem

```
CREATE TABLE hempel.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';
```
-- ilike none case-sensitive LIKE

-- Dodanie serial primary key

```
ALTER TABLE hempel.intersects
ADD COLUMN rid SERIAL PRIMARY KEY;
```
-- SERIAL datatype allows automatically generate unique int numbers

-- Utworzenie indeksu przestrzennego

```
CREATE INDEX idx_intersects_rast_gist ON hempel.intersects
USING gist (ST_ConvexHull(rast));
```
-- ST_ConvexHull - oblicza otoczkę wypukłą geometrii

-- Dodanie raster constraints

-- schema::name table_name::name raster_column::name

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

-- Przykład 2. - ST_Clip

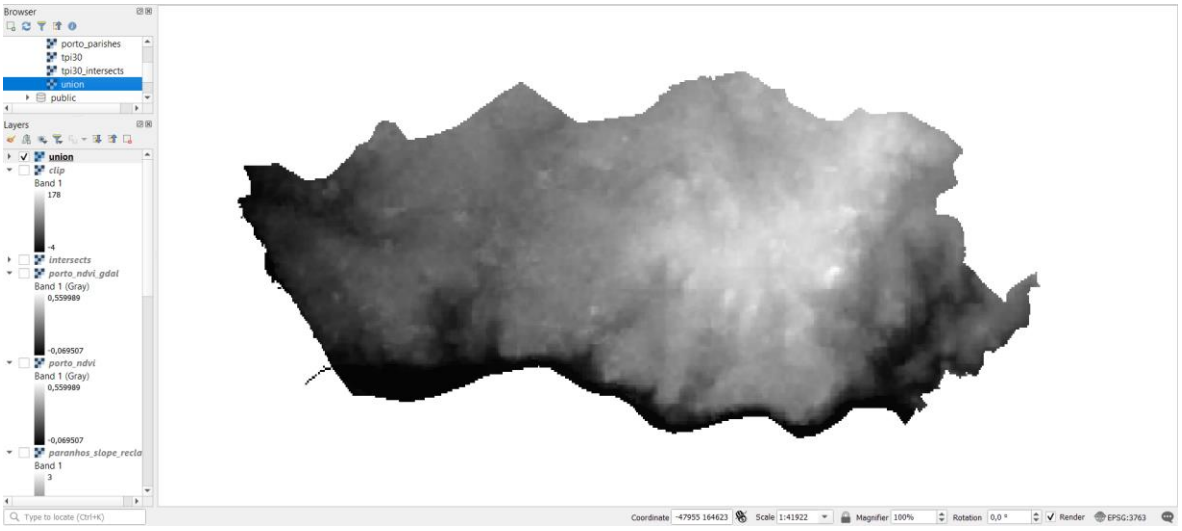
-- Obcinanie rastra na podstawie wektora

```
CREATE TABLE hempel.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 hempel.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

-- konwertuje geometry na raster

```
CREATE TABLE hempel.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';
```

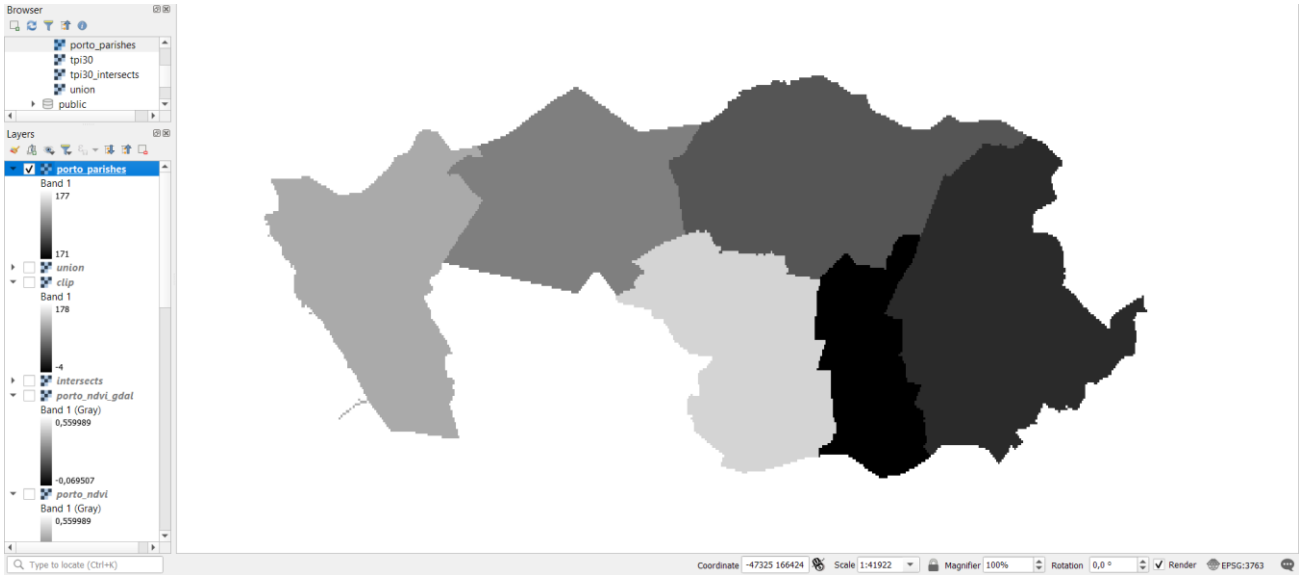
-- Przykład 2. - ST_Union

```
DROP TABLE hempel.porto_parishes; --> drop table porto_parishes first
CREATE TABLE hempel.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

-- returns a set of rasters resulting from the split of the input raster

```
DROP TABLE hempel.porto_parishes; --> drop table porto_parishes first
CREATE TABLE hempel.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';
```



KONWERTOWANIE RASTRÓW NA WEKTORY (wektoryzowanie)

-- Przykład 1. - ST_Intersection

```
CREATE TABLE hempel.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

-- konwertuje rastry na wektory (poligony)

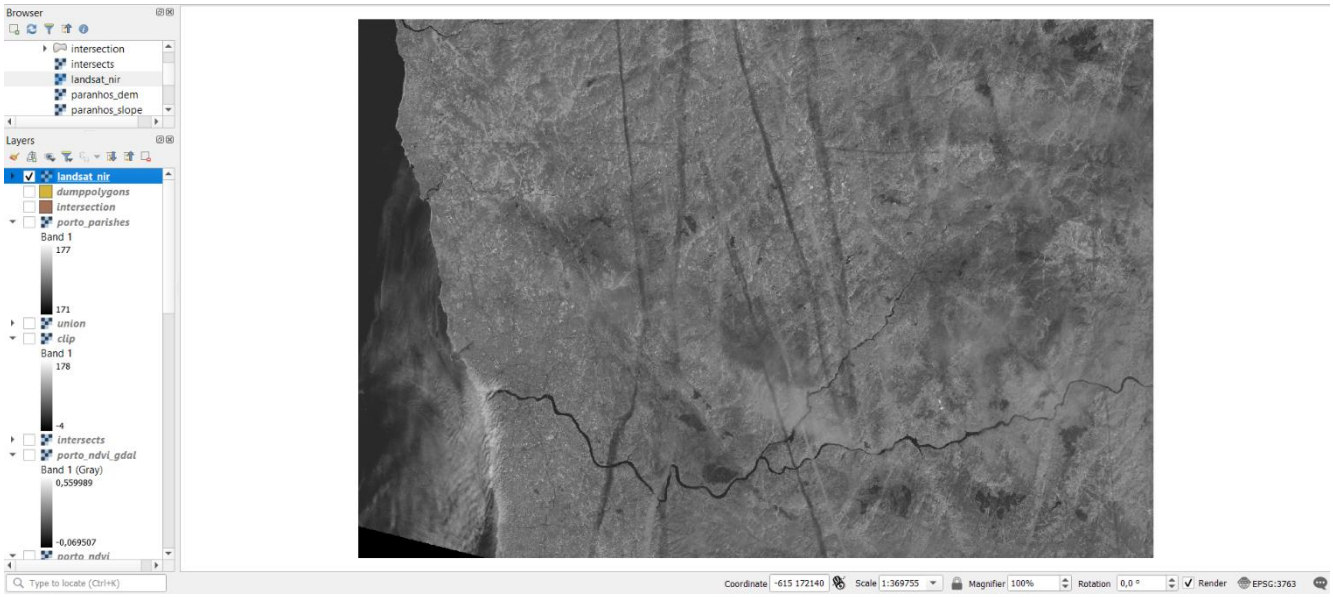
```
CREATE TABLE hempel.dumppolygons AS
SELECT
a.rid,(ST_DumpAsPolygons(ST_Clip(a.rast,b.geom))).geom,(ST_DumpAsPolygons(ST_Clip(
a.rast,b.geom))).val
FROM rasters.landsat8 AS a, vectors.porto_parishes AS b
WHERE b.parish ilike 'paranhos' and ST_Intersects(b.geom,a.rast);
```

ANALIZA RASTRÓW

-- Przykład 1. - ST_Band

-- returns one or more bands of an existing raster as a new raster

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

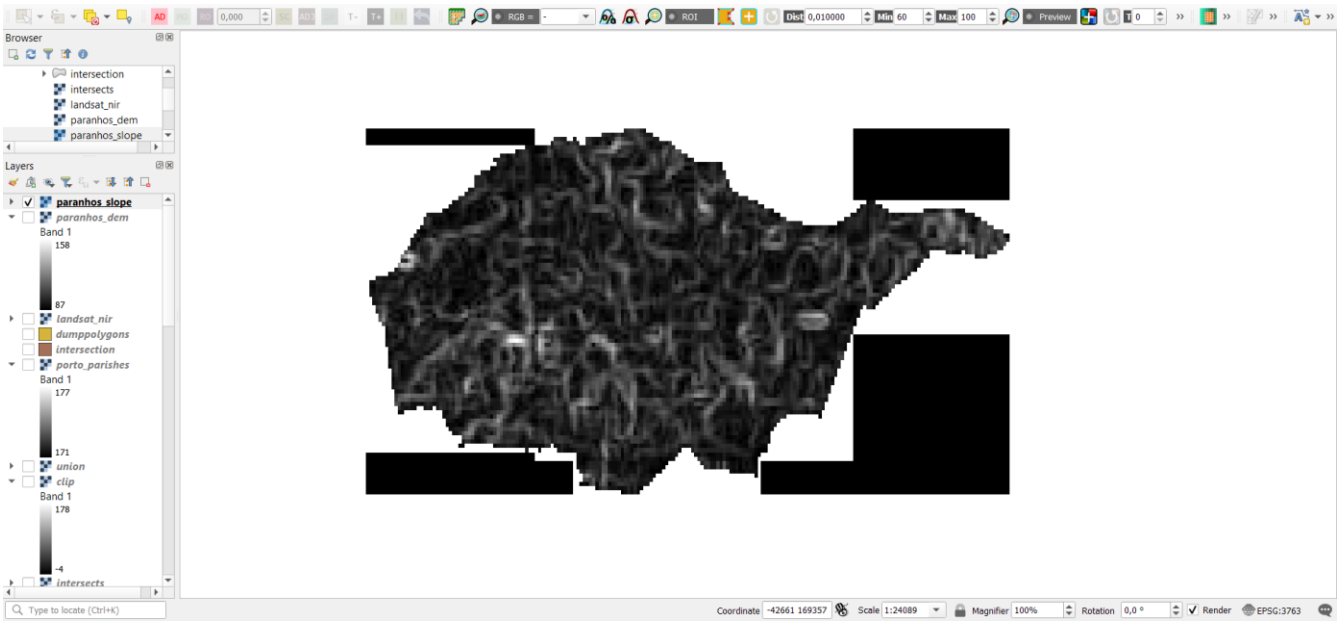


-- Przykład 2. - ST_Clip

CREATE TABLE hempel.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 - generowanie nachylenia

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



-- Przykład 4. ST_Reclass - reklasyfikacja rastra

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

-- Przykład 5. - ST_SummaryStats - obliczanie statystyk rastra

```
SELECT ST_SummaryStats(a.rast) AS stats
FROM hempel.paranhos_dem AS a;
```

-- Przykład 6. - ST_SummaryStats oraz Union

```
SELECT ST_SummaryStats(ST_Union(a.rast))
FROM hempel.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 hempel.paranhos_dem AS a
)
SELECT (stats).min,(stats).max,(stats).mean FROM t;
```

Data Output | Explain | Messages | Notifications

	min double precision	max double precision	mean double precision
1	87	158	122.52731281948482

-- Przykład 8 - ST_SummaryStats w połączeniu z 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;
```

	parish character varying (254)	min double precision	max double precision	mean double precision
1	Bonfim	1	159	107.5658842667906
2	Campanhã	0	178	74.66732213085449
3	Paranhos	87	158	122.52731281948482
4	Ramalde	48	108	77.58444444444444
5	União das freguesias de Aldoar, Foz do Douro e Nevogilde	-4	83	34.66735489791237
6	União das freguesias de Cedofeita, Santo Ildefonso, Sé, Miragaia, São Nicolau e Vitória	1	157	95.00277741039545
7	União das freguesias de Lordelo do Ouro e Massarelos	-1	117	49.50051440329218

-- Przykład 9 - ST_Value

-- returns the value of given band in a given column

```
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;
```

	name character varying (48)	st_value double precision
1	Aldeia São Miguel	96
2	Alpendurada e Matos	145
3	Amarante	71
4	Baião	581
5	Cabeceiras de Basto	[null]
6	Castelo de Paiva	284
7	Celorico de Basto	227
8	Cinfães	405
9	Espinho	14
10	Fafe	338
11	Fajozes	53
12	Felgueiras	320
13	Gondomar	123
14	Guifões	69
15	Guimarães	197
16	Lousada	289
17	Maia	111
18	Marco de Canaveses	193
19	Matosinhos	29
20	Paços de Ferreira	300
21	Paredes	178
22	Penafiel	281
23	Porto	81
24	Póvoa de Varzim	15
25	Rio do Moinhos	106
26	São Mamede de Infesta	97
27	Torrão	89
28	Trofa	32
29	Valongo	139
30	Vila do Conde	12
31	Vila Nova de Famalicão	116
32	Vila Nova de Gaia	82
33	Vizela	156

TOPOGRAPHIC POSITION INDEX (TPI)

-- Przykład 10 - ST_TPI

```
CREATE TABLE hempel.tpi30 as
select ST_TPI(a.rast,1) as rast
from rasters.dem a;
```

-- Stworzenie indeksu przestrzennego

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

-- Dodanie constraintów

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

-- Problem do samodzielnego zrealizowania

-- ograniczenie obszaru do mniejszego regionu

```
CREATE TABLE hempel.tpi30_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';


SELECT * FROM hempel.tpi30_intersects;
```

-- Tworzenie indeksu przestrzennego

```
CREATE INDEX idx_tpi30_rast_gist_intersects ON hempel.tpi30_intersects
USING gist (ST_ConvexHull(rast));
```

-- Dodanie constraintów

```
SELECT AddRasterConstraints('hempel'::name, 'tpi30_intersects'::name,'rast'::name);
```

Porównanie czasów wykonania zapytań

	tpi30	tpi30_intersects
Tworzenie tabeli	37 s	9 s 315 ms
Tworzenie indeksu	52 ms	34 ms
Dodanie constraintu	214 ms	105 ms

ALGEBRA MAP

-- Przykład 1. - Wyrażenie Algebry Map

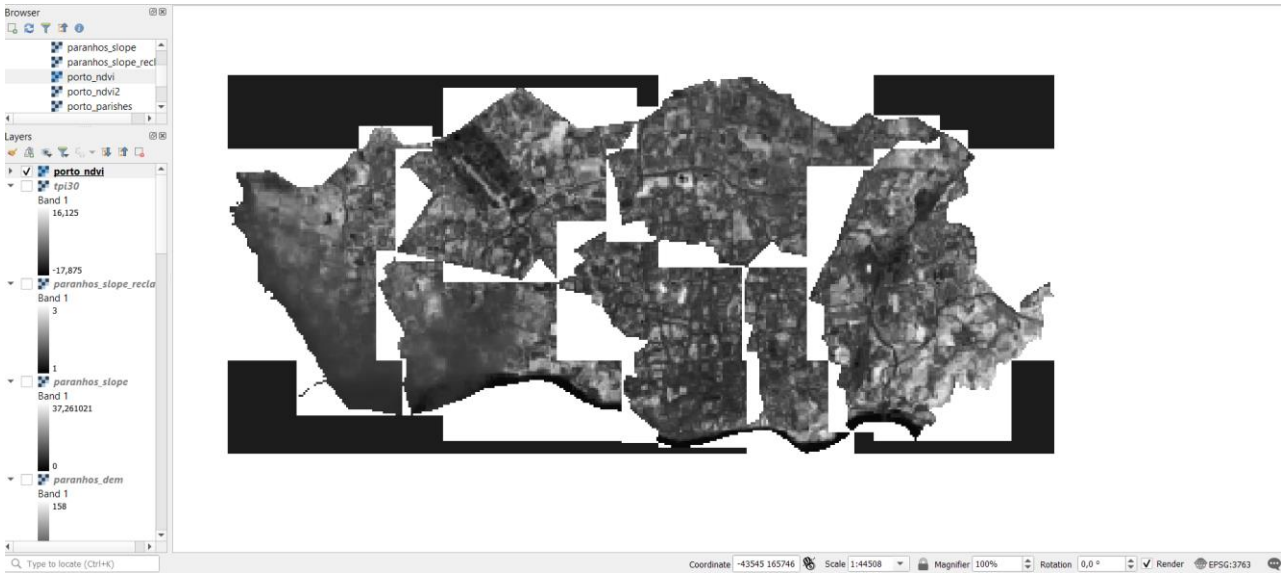
```
CREATE TABLE hempel.porto_ndvi AS
WITH r AS (
	SELECT a.rid, ST_Clip(a.rast, b.geom,true) AS rast
	FROM rasters.landsat8 AS a, vectors.porto_parishes AS b
	WHERE b.municipality ilike 'porto' and ST_Intersects(b.geom,a.rast)
)
SELECT
	r.rid,ST_MapAlgebra(
	r.rast, 1,
	r.rast, 4,
		'([rast2.val] -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

-- Tworzenie indeksu przestrzennego

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

-- Dodanie constraintów

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



-- Przykład 2. - Funkcja zwrotna

```
create or replace function hempel.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;

CREATE TABLE hempel.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],
	'hempel.ndvi(double precision[],
integer[],text[])'::regprocedure, --> This is the function!
'32BF'::text
) AS rast
FROM r;
```

-- Tworzenie indeksu przestrzennego

```
CREATE INDEX idx_porto_ndvi2_rast_gist ON hempel.porto_ndvi2
USING gist (ST_ConvexHull(rast));
```

-- Dodanie constraintów

```
SELECT AddRasterConstraints('hempel'::name, 'porto_ndvi2'::name,'rast'::name);
```

EKSPORT DANYCH

-- Przykład 0 - Użycie QGIS

```
CREATE TABLE hempel.porto_ndvi_qgis AS
SELECT ST_Union(rast) FROM hempel.porto_ndvi;
```

-- Przykład 1 - ST_AsTiff

```
SELECT ST_AsTiff(ST_Union(rast))
FROM hempel.porto_ndvi;
```

-- Przykład 2 - ST_AsGDALRaster

```
SELECT ST_AsGDALRaster(ST_Union(rast), 'GTiff', ARRAY['COMPRESS=DEFLATE',
'PREDICTOR=2', 'PZLEVEL=9'])
FROM hempel.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 hempel.porto_ndvi;
----------------------------------------------
SELECT lo_export(loid, 'D:\porto_ndvi.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 (Geospatial Data Abstraction Library)

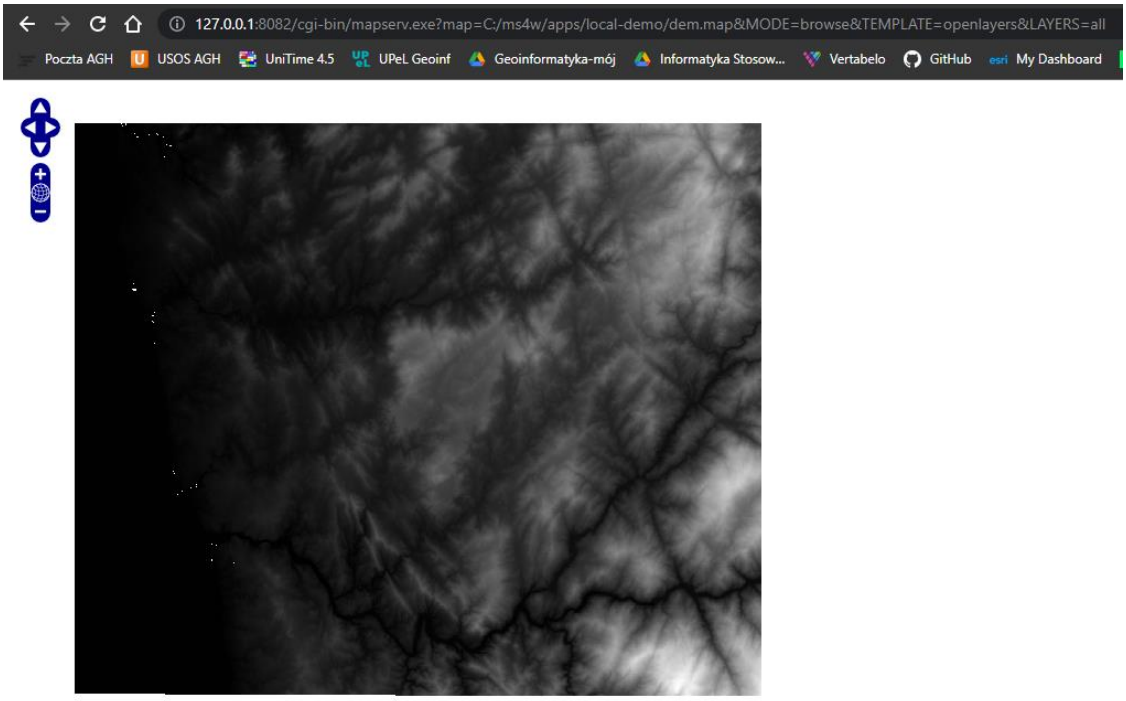
Uruchomienie gdal:

- conda create -n gdal
- activate gdal
- conda install -c conda-forge gdal

gdal_translate -co COMPRESS=DEFLATE -co PREDICTOR=2 -co ZLEVEL=9 PG:"host=localhost port=5432 dbname=cwiczenia6 user=postgres password=Siatka07 schema=hempel table=porto_ndvi mode=2" "C:\Users\kubah\Desktop\AGH zajecia\Bazy danych przestrzennych laborki\Cwiczenia_6\rasters\porto_ndvi_gdal.tiff"

PUBLIKOWANIE DANYCH ZA POMOCĄ MAPSERVER

http://127.0.0.1:8082/cgi-bin/mapserv.exe?map=C:/ms4w/apps/local-demo/dem.map&MODE=browse&TEMPLATE=openlayers&LAYERS=all



-- Zawartość pliku dem.map

```
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=cwiczenia6 user=postgres
password=Siatka07 schema=rasters table=dem mode=2"
      PROCESSING "SCALE=AUTO"
      PROCESSING "NODATA=-32767"
      OFFSITE 0 0 0
      METADATA
         'wms_title' 'srtm'
      END
   END
END
```

-- GEOSERVER

```
CREATE TABLE public.mosaic (
   name character varying(254) COLLATE pg_catalog."default" NOT NULL,
   tiletable character varying(254) COLLATE pg_catalog."default" NOT NULL,
   minx double precision,
   miny double precision,
   maxx double precision,
   maxy double precision,
   resx double precision,
```

```
    resy double precision,
    CONSTRAINT mosaic_pkey PRIMARY KEY (name, tiletable)
);
insert into mosaic (name,tiletable) values ('mosaicpgraster','rasters.dem');

SELECT * FROM public.mosaic;
```

-- Zawartość pliku *mosaicpgraster.pgraster.xml*

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE ImageMosaicJDBCConfig [
    <!ENTITY mapping PUBLIC "mapping"  "mapping.pgraster.xml.inc">
    <!ENTITY connect PUBLIC "connect"  "connect.pgraster.xml.inc">
]>

<config version="1.0">
    <coverageName name="mosaicpgraster"/>
    <coordsys name="EPSG:3763"/>
    <scaleop  interpolation="1"/>
    <axisOrder ignore="false"/>
    &mapping;
    &connect;
</config>
```

-- Zawartość pliku *mapping.pgraster.xml.inc*

```
<spatialExtension name="pgraster"/>
 <mapping>
    <masterTable name="mosaic" >
            <coverageNameAttribute name="name"/>
            <maxXAttribute name="maxX"/>
            <maxYAttribute name="maxY"/>
            <minXAttribute name="minX"/>
            <minYAttribute name="minY"/>
            <resXAttribute name="resX"/>
            <resYAttribute name="resY"/>
            <tileTableNameAtribute name="tiletable" />
    </masterTable>
    <tileTable>
            <blobAttributeName name="rast" />
    </tileTable>
</mapping>
```

-- Zawartość pliku *connect.pgraster.xml.inc*

```
<connect>
  <dstype value="DBCP"/>
  <username value="postgres"/>
  <password value="Siatka07"/>
  <jdbcUrl value="jdbc:postgresql://localhost:5432/cwiczenia6"/>
  <driverClassName value="org.postgresql.Driver"/>
  <maxActive value="10"/>
  <maxIdle value="0"/>
</connect>
```

http://localhost:9000/geoserver/bdp_rasters/wms?service=WMS&version=1.1.0&request=GetMap&layers=bdp_rasters%3Amosaicpgraster&bbox=-58968.422190841186%2C147735.24369472192%2C13425.075426302268%2C206234.677340203 95&width=768&height=620&srs=EPSG%3A3763&styles=&format=application/openlayers