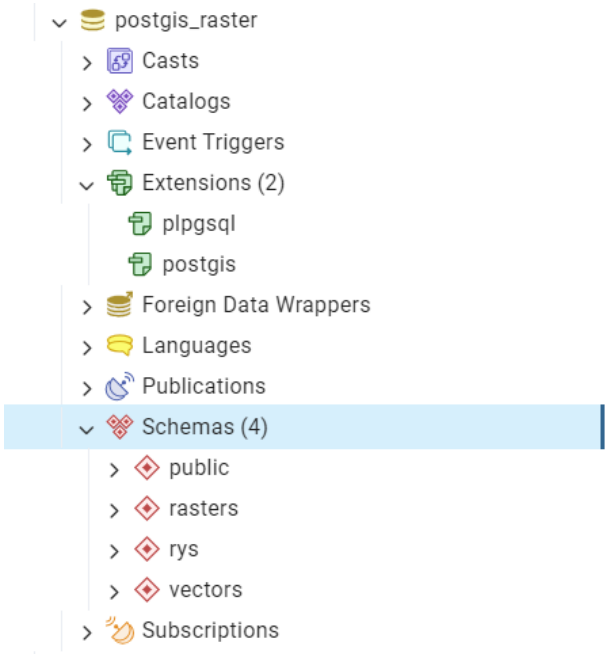
Bazy danych przestrzennych – ćwiczenia z PostGIS raster

Do wykonania poniższych poleceń użyto ***pgAdmin 4*** oraz ***QGIS 3.28.11.***

1. Utworzono nową bazę danych ***postgis_raster*** oraz załadowano do niej kopię bazy danej z pliku ***postgis_raster.backup.*** W tym celu w wierszu poleceń wpisano:

   ```
   pg_restore -h localhost -U postgres -d postgis_raster -clean "C:/[ścieżka]/PostGIS/postgis_raster.backup"
   ```

   Zmieniono nazwę schematu na swoje nazwisko: `ALTER SCHEMA schema_name RENAME TO rys;`

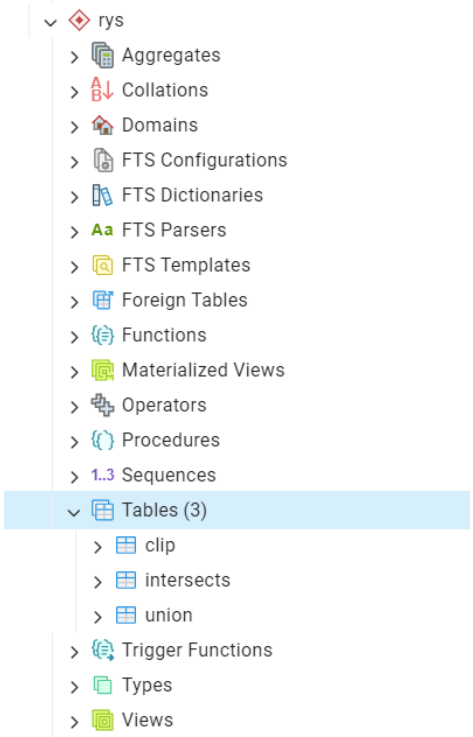


2. Załadowana pliki rastrowe ***Landsat8_L1TP_RGBN.tif*** oraz ***srtm_1arc_v3.tif.***

   ```
   raster2pgsql.exe -s 3763 -N -32767 -t 128x1028 -I -C -M -d "C:\RYS\Desktop\Landsat8_L1TP_RGBN.TIF" rasters.landsat8 | psql -d postgis_raster -h localhost -U postgres -p 5432

   raster2pgsql.exe -s 3763 -N -32767 -t 100x100 -I -C -M -d "C:\RYS\Desktop\srtm_1arc_v3.TIF" rasters.dem | psql -d postgis_raster -h localhost -U postgres -p 5432
   ```

3. **Tworzenie rastrów z już istniejących rastrów i interakcja z wektorami**

```
--Przykład 1 - ST_Intersects

--przycięcie rastra z wektorem
CREATE TABLE rys.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';

select * from rys.intersects

--dodanie serial primary key:
alter table rys.intersects
add column rid SERIAL PRIMARY KEY;
--utworzenie indeksu przestrzennego:
CREATE INDEX idx_intersects_rast_gist ON rys.intersects
USING gist (ST_ConvexHull(rast));
--dodanie raster constraints:
SELECT AddRasterConstraints('rys'::name,
'intersects'::name,'rast'::name);
--Przykład 2 - ST_Clip
--Obcinanie rastra na podstawie wektora.
CREATE TABLE rys.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 rys.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);
```



4. Tworzenie rastrów z wektorów (rastrowanie)

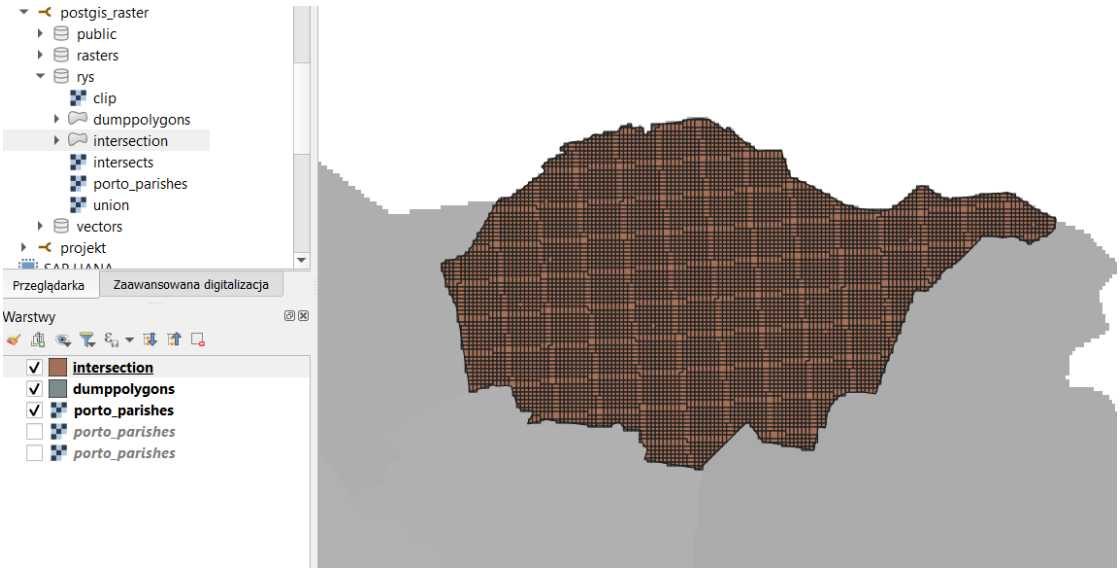
```
--Przykład 1 - ST_AsRaster
CREATE TABLE rys.porto_parishes AS
WITH r AS (
SELECT rast FROM rasters.dem
LIMIT 1
)
SELECT ST_AsRaster(a.geom,r.rast,'8BUI',a.id,-32767) AS rast
FROM vectors.porto_parishes AS a, r
WHERE a.municipality ilike 'porto';
```

```
--Przykład 2 - ST_Union
DROP TABLE rys.porto_parishes; --> drop table porto_parishes first
CREATE TABLE rys.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
DROP TABLE rys.porto_parishes; --> drop table porto_parishes first
CREATE TABLE rys.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';
```

5. Konwertowanie rastrów na wektory (wektoryzowanie)

```
--Przykład 1 - ST_Intersection
create table rys.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
CREATE TABLE rys.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);
```



6. Analiza rastrów

```
--Przykład 1 - ST_Band
CREATE TABLE rys.landsat_nir AS
SELECT rid, ST_Band(rast,4) AS rast
FROM rasters.landsat8;

--Przykład 2 - ST_Clip
CREATE TABLE rys.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
CREATE TABLE rys.paranhos_slope AS
SELECT a.rid,ST_Slope(a.rast,1,'32BF','PERCENTAGE') as rast
FROM rys.paranhos_dem AS a;

--Przykład 4 - ST_Reclass
CREATE TABLE rys.paranhos_slope_reclass AS
SELECT a.rid,ST_Reclass(a.rast,1,']0-15]:1, (15-30]:2, (30-9999:3',
'32BF',0)
FROM rys.paranhos_slope AS a;
```

```
--Przykład 5 - ST_SummaryStats
SELECT st_summarystats(a.rast) AS stats
FROM rys.paranhos_dem AS a;
```

Data Output | Messages | Graph Visualiser | Notifications

	stats summarystats
1	(2616,278385,106.41628440366972,11.622628762211638,87,14…
2	(682,95581,140.14809384164224,12.078072186605759,103,158)
3	(216,31874,147.5648148148148,4.262830628315728,137,158)
4	(6463,816615,126.35231316725978,14.0438229209133,94,158)

```
--Przykład 6 - ST_SummaryStats oraz Union
SELECT st_summarystats(ST_Union(a.rast))
FROM rys.paranhos_dem AS a;
```

Data Output | Messages | Graph Visualiser | Notifications

	st_summarystats summarystats
1	(9977,1222455,122.52731281948482,16.908004202736272,87,15…

```
--Przykład 7 - ST_SummaryStats z lepszą kontrolą złożonego typu danych
WITH t AS (
SELECT st_summarystats(ST_Union(a.rast)) AS stats
FROM rys.paranhos_dem AS a
)
SELECT (stats).min,(stats).max,(stats).mean FROM t;
```

Data Output | Messages | Graph Visualiser | 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;
```

Data Output | Messages | Graph Visualiser | Notifications

	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 Vitó…	1	157	95.00277741039545
7	União das freguesias de Lordelo do Ouro e Massarelos	-1	117	49.50051440329218

```
--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;
```

Data Output | Messages | Graph Visualiser | Notifications

	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

7. Topographic Position Index (TPI)

```
148  --Przykład 10 - ST_TPI
149  create table rys.tpi30 as
150  select ST_TPI(a.rast,1) as rast
151  from rasters.dem a;
152
```

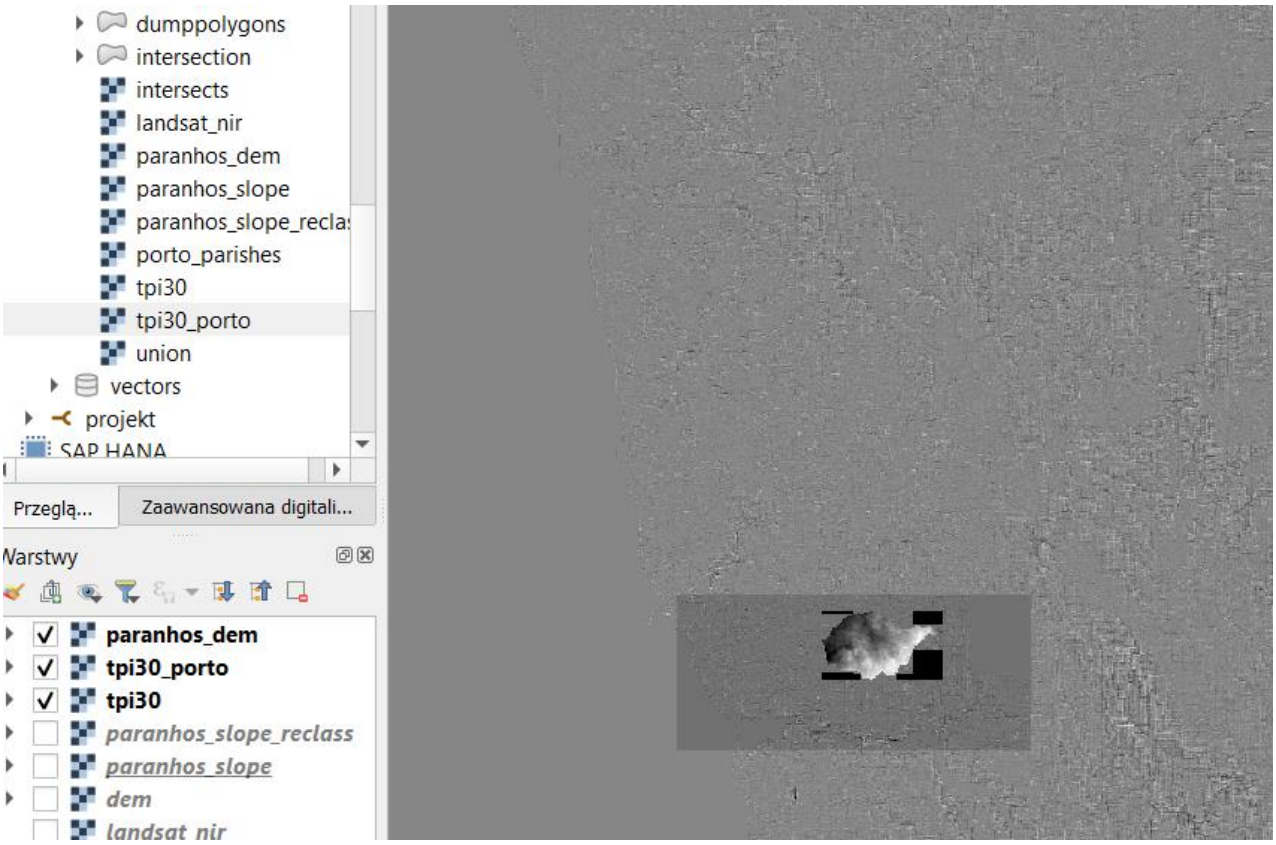
Data Output | Messages | Graph Visualiser | Notificatior

```
SELECT 589

Query returned successfully in 37 secs 583 msec.
```

```
153  create table rys.tpi30_porto as
154  select ST_TPI(a.rast,1) as rast
155  from rasters.dem AS a, vectors.porto_parishes AS b
156  WHERE ST_Intersects(a.rast, b.geom) AND b.municipality ilike 'porto';
157
```

Data Output | Messages | Graph Visualiser | Notifications

```
SELECT 25

Query returned successfully in 1 secs 614 msec.
```



8. Algebra map

```
--Przykład 1 - Wyrażenie Algebry Map
CREATE TABLE rys.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;

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

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

```
--Przykład 2 - Funkcja zwrotna
create or replace function rys.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 rys.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],
'rys.ndvi(double precision[],
integer[],text[])'::regprocedure, --> This is the function!
'32BF'::text
) AS rast
FROM r;

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

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

9. Eksport danych

```
--Eksport danych
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 rys.porto_ndvi;
----------------------------------------------
SELECT lo_export(loid, 'C:\RYS\Desktop\porto_ndvi.tiff')
FROM tmp_out;
----------------------------------------------
SELECT lo_unlink(loid)
FROM tmp_out;
```