1. Algebra map

- Wyrażenie algebry map

```sql
1   CREATE TABLE pelc.porto_ndvi AS
2   WITH r AS (
3   SELECT a.rid,ST_Clip(a.rast, b.geom,true) AS rast
4   FROM rasters.landsat8 AS a, vectors.porto_parishes AS b
5   WHERE b.municipality ilike 'porto' and ST_Intersects(b.geom,a.rast)
6   )
7   SELECT
8   r.rid,ST_MapAlgebra(
9   r.rast, 1,
10  r.rast, 4,
11  '([rast2.val] - [rast1.val]) / ([rast2.val] +
12  [rast1.val])::float','32BF'
13  ) AS rast
14  FROM r;
15
```

Data Output   Messages   Notifications

```
SELECT 23

Query returned successfully in 490 msec.
```

```sql
16  CREATE INDEX idx_porto_ndvi_rast_gist ON pelc.porto_ndvi
17  USING gist (ST_ConvexHull(rast));
18
```

Data Output   Messages   Notifications

```
CREATE INDEX

Query returned successfully in 72 msec.
```

```
18
19
20   SELECT AddRasterConstraints('pelc'::name,
21   'porto_ndvi'::name,'rast'::name);
22
23
24
```

Data Output   Messages   Notifications

addrasterconstraints 🔒 boolean
1   true

- Funkcja zwrotna

```
1   create or replace function pelc.ndvi(
2   value double precision [] [] [],
3   pos integer [][],
4   VARIADIC userargs text []
5   )
6   RETURNS double precision AS
7   $$
8 ▼ BEGIN
9   --RAISE NOTICE 'Pixel Value: %', value [1][1][1];-->For debug purposes
10  RETURN (value [2][1][1] - value [1][1][1])/(value [2][1][1]+value
11  [1][1][1]); --> NDVI calculation!
12  END;
13  $$
14  LANGUAGE 'plpgsql' IMMUTABLE COST 1000;
15
16
17
```

Data Output   Messages   Notifications

CREATE FUNCTION

Query returned successfully in 71 msec.

```
21
22  CREATE TABLE pelc.porto_ndvi2 AS
23  WITH r AS (
24  SELECT a.rid,ST_Clip(a.rast, b.geom,true) AS rast
25  FROM rasters.landsat8 AS a, vectors.porto_parishes AS b
26  WHERE b.municipality ilike 'porto' and ST_Intersects(b.geom,a.rast)
27  )
28  SELECT
29  r.rid,ST_MapAlgebra(
30  r.rast, ARRAY[1,4],
31  'pelc.ndvi(double precision[],
32  integer[],text[])'::regprocedure, --> This is the function!
33  '32BF'::text
34  ) AS rast
35  FROM r;
36
37
38
```

Data Output   Messages   Notifications

```
SELECT 23

Query returned successfully in 176 msec.
```

```
37
38  CREATE INDEX idx_porto_ndvi2_rast_gist ON pelc.porto_ndvi2
39  USING gist (ST_ConvexHull(rast));
40
41
```

Data Output   Messages   Notifications

```
CREATE INDEX

Query returned successfully in 38 msec.
```

```
42
43  SELECT AddRasterConstraints('pelc'::name,
44  'porto_ndvi2'::name,'rast'::name);
45
```

Data Output   Messages   Notifications

addrasterconstraints 🔒 boolean
1	true

2. Funkcje TPI

- ST_TPI

Synopsis

raster **ST_TPI**(raster *rast*, integer *nband*, raster *customextent*, text *pixeltype="32BF"* , boolean *interpolate_nodata=FALSE*);
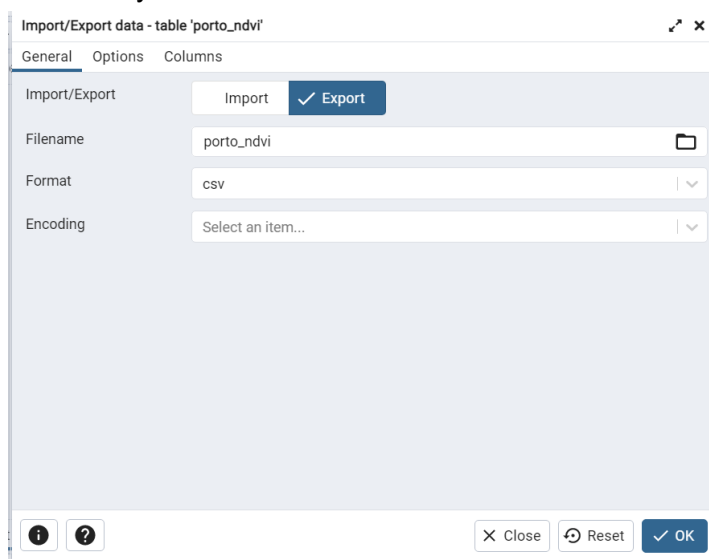
Description

Calculates the Topographic Position Index, which is defined as the focal mean with radius of one minus the center cell.

- _st_tpi4ma

Option	Value
Returns	double precision
Language	plpgsql
Parameters	value double precision []
	pos integer []
	variadic userargs text [] = NULL::text[]

3. Eksport danych

- Użycie QGIS

- ST_AsTiff

```
2  SELECT ST_AsTiff(ST_Union(rast))
3  FROM pelc.porto_ndvi;
4
5
```

Data Output | Messages | Notifications

st_astiff 🔒 bytea
1 [binary dat...

- ST_AsGDALRaster

```
1
2  SELECT ST_AsGDALRaster(ST_Union(rast), 'GTiff', ARRAY['COMPRESS=DEFLATE',
3  'PREDICTOR=2', 'PZLEVEL=9'])
4  FROM pelc.porto_ndvi;
5
```

Data Output | Messages | Notifications

st_asgdalraster 🔒 bytea
1 [binary data]

```
7  SELECT ST_GDALDrivers();
8
```

Data Output | Messages | Notifications

	st_gdaldrivers record
1	(0,GTiff,GeoTIFF,t,t,"<CreationOptionList> <Option name='COMPRESS' type='string-select'> <Value>NONE</Value> <Value>LZW</Value> <Value>PACKBITS</Value> <Value>JPEG</Value> <Value
2	(1,AAIGrid,"Arc/Info ASCII Grid",t,t,"<CreationOptionList>
3	(2,DTED,"DTED Elevation Raster",t,t,"")
4	(3,PNG,"Portable Network Graphics",t,t,"<CreationOptionList>
5	(4,JPEG,"JPEG JFIF",t,t,"<CreationOptionList>
6	(5,GIF,"Graphics Interchange Format (.gif)",t,t,"<CreationOptionList>
7	(6,USGSDEM,"USGS Optional ASCII DEM (and CDED)",t,t,"<CreationOptionList> <Option name='PRODUCT' type='string-select' description='Specific Product Type'> <Value>DEFAULT</Value> <Value>CDED5(
8	(7,XYZ,"ASCII Gridded XYZ",t,t,"<CreationOptionList> <Option name='COLUMN_SEPARATOR' type='string' default=' ' description='Separator between fields.'/> <Option name='ADD_HEADER_LINE' type='boolean' de

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

```
 3  CREATE TABLE tmp_out AS
 4  SELECT lo_from_bytea(0,
 5   ST_AsGDALRaster(ST_Union(rast), 'GTiff', ARRAY['COMPRESS=DEFLATE',
 6  'PREDICTOR=2', 'PZLEVEL=9'])
 7   ) AS loid
 8  FROM pelc.porto_ndvi;
 9  ------------------------------------------
10  SELECT lo_export(loid, 'C:\cw3-bazy\myraster.tiff') --> Save the file in a place
11  -- where the user postgres have access. In windows a flash drive usualy works fine.
12   FROM tmp_out;
13  ------------------------------------------
14  SELECT lo_unlink(loid)
15   FROM tmp_out; --> Delete the large object.
16
17
```

Data Output   Messages   Notifications

	lo_unlink integer 🔒
1	1

- Użycie Gdal

```
C:\Users\Ewa>gdal_translate -co COMPRESS=DEFLATE -co PREDICTOR=2 -co ZLEVEL=9 PG:"host=localhost port=5432 dbname=cw6 user=postgres password=test123 schema=pelc table=porto_ndvi mode=2" porto_ndvi.tiff
'gdal_translate' is not recognized as an internal or external command,
operable program or batch file.
```

4. Opublikowanie mapy za pomocą MapServer

- W mapfile.map jako hosta podaję adres IP WSL-a (dostępny z terminala po wpisaniu komendy 'ipconfig')

```
Ethernet adapter vEthernet (WSL):

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::8f5e:f2f:9433:3f9c%51
   IPv4 Address. . . . . . . . . . . : 192.168.160.1
   Subnet Mask . . . . . . . . . . . : 255.255.240.0
   Default Gateway . . . . . . . . . :
```

- Modyfikuję mapfile.map, aby odpowiadał moim danym

```
mapfile.map
 1   MAP
 2       NAME 'map'
 3       SIZE 800 650
 4       STATUS ON
 5       EXTENT -58968 145487 30916 206234
 6       UNITS METERS
 7       WEB
 8           METADATA
 9               'wms_title' 'Terrain wms'
10               'wms_srs' 'EPSG:3763 EPSG:4326 EPSG:3857'
11               'wms_enable_request' '*'
12               'wms_onlineresource'
13               'http://54.37.13.53/mapservices/srtm'
14           END
15       END
16       PROJECTION
17           'init=epsg:3763'
18       END
19       LAYER
20           NAME srtm
21           TYPE raster
22           STATUS OFF
23           DATA "PG:host=192.168.160.1 port=5432 dbname='cw6'
24           user='postgres' password='test123' schema='rasters' table='dem' mode='2'"
25           PROCESSING "SCALE=AUTO"
26           PROCESSING "NODATA=-32767"
27           OFFSITE 0 0 0
28           METADATA
29               'wms_title' 'srtm'
30           END
31       END
32   END
33
```

- uruchamiam dockera oraz łączę się z bazą danych

```
root@a7c8fd964d59:/# psql postgres://postgres:test123@192.168.160.1/cw6
psql (16.1 (Ubuntu 16.1-1.pgdg20.04+1), server 14.7)
Type "help" for help.

cw6=#
```

- Wpisuje adres podany w tutorialu