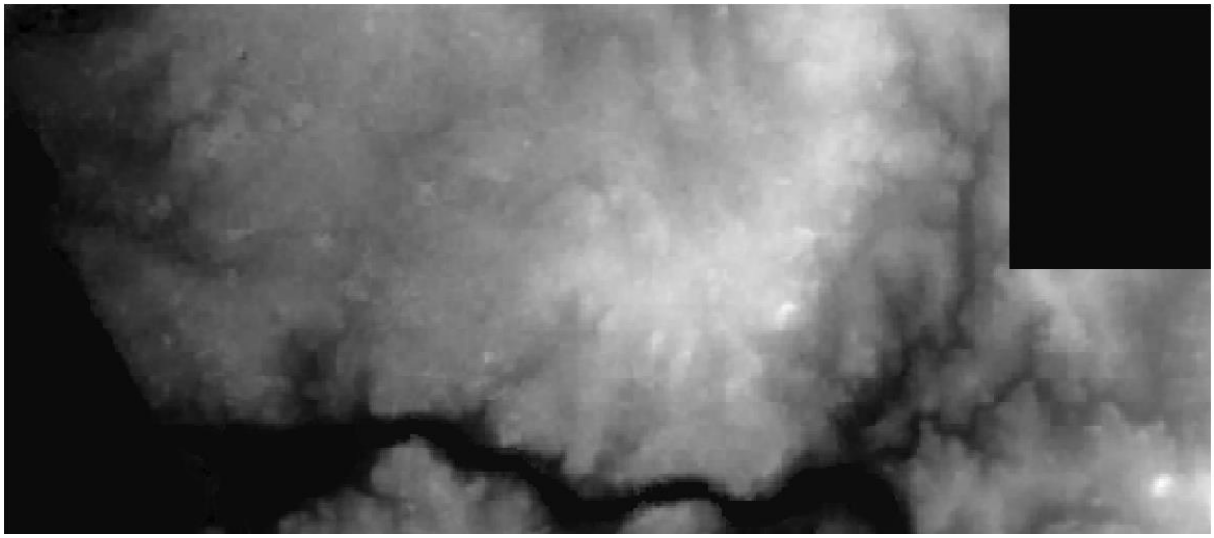
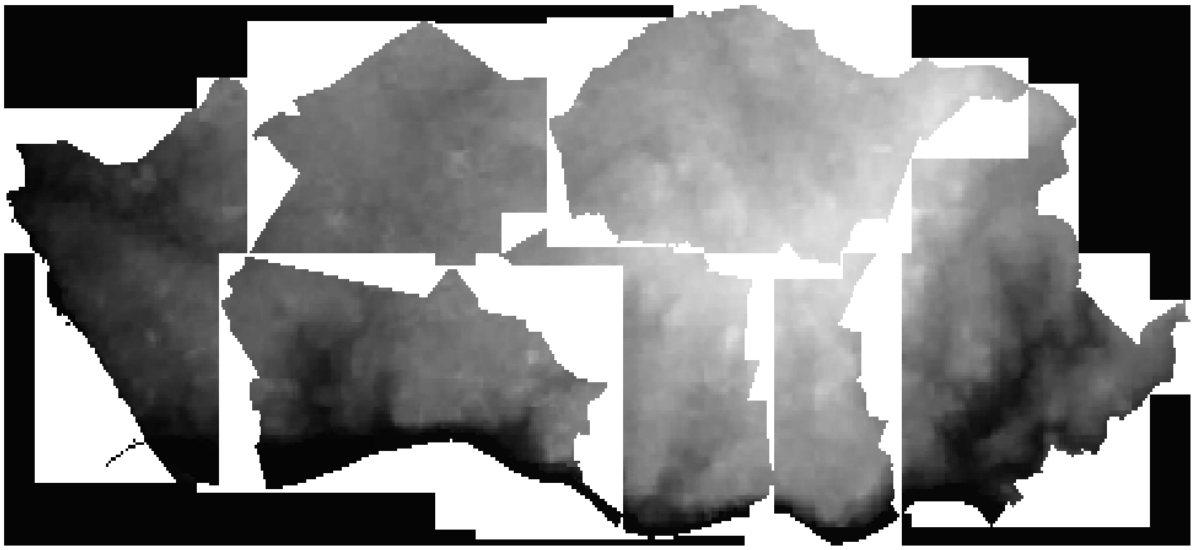
```
--Przykład 1 - ST_Intersects
--Przecięcie rastra z wektorem.

create schema kwasniak;

CREATE TABLE kwasniak.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 kwasniak.intersects
add column rid SERIAL PRIMARY KEY;
--2. utworzenie indeksu przestrzennego:
CREATE INDEX idx_intersects_rast_gist ON kwasniak.intersects
USING gist (ST_ConvexHull(rast));

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



--Przykład 2 - ST_Clip

--Obcinanie rastra na podstawie wektora.

```
CREATE TABLE kwasniak.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';
```

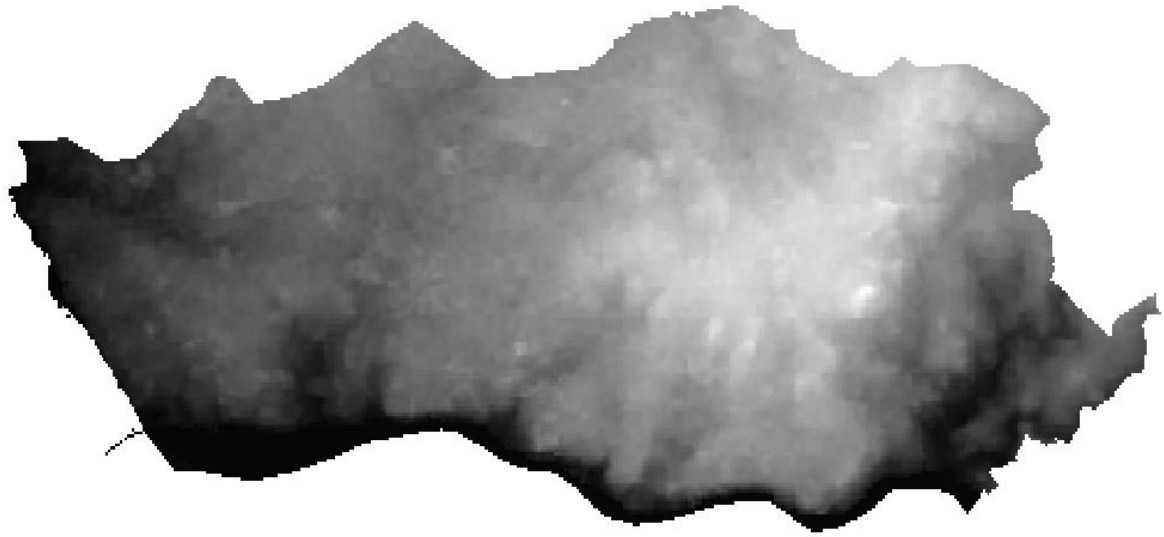


--Przykład 3 - ST_Union

--Połączenie wielu kafelków w jeden raster.

```
CREATE TABLE kwasniak.union AS
SELECT ST_Union(ST_Clip(a.rast, b.geom, true))
FROM rasters.dem AS a, vectors.porto_parishes AS b
```

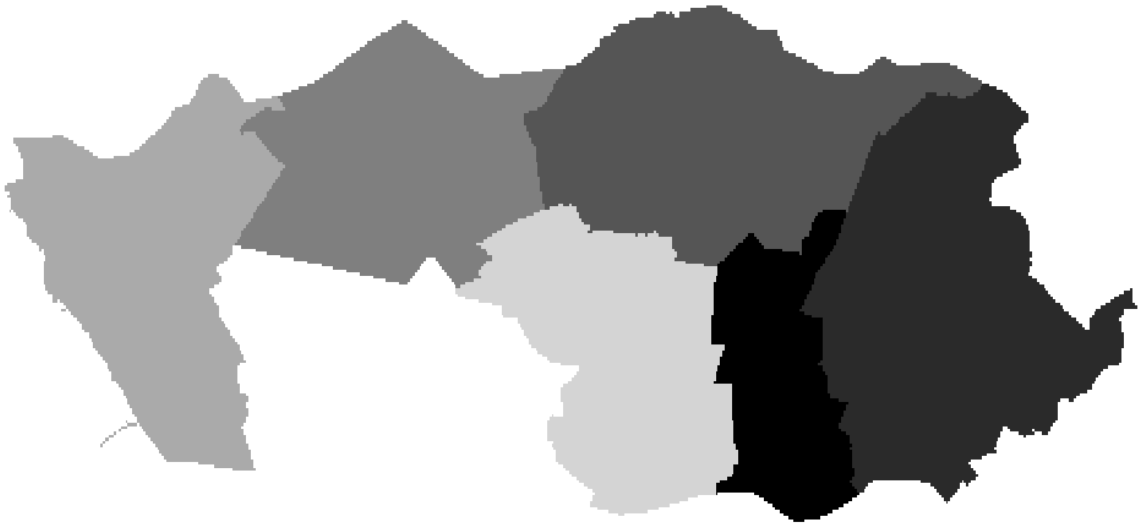
```
WHERE b.municipality ilike 'porto' and ST_Intersects(b.geom,a.rast);


--select * from kwasniak.union;
```



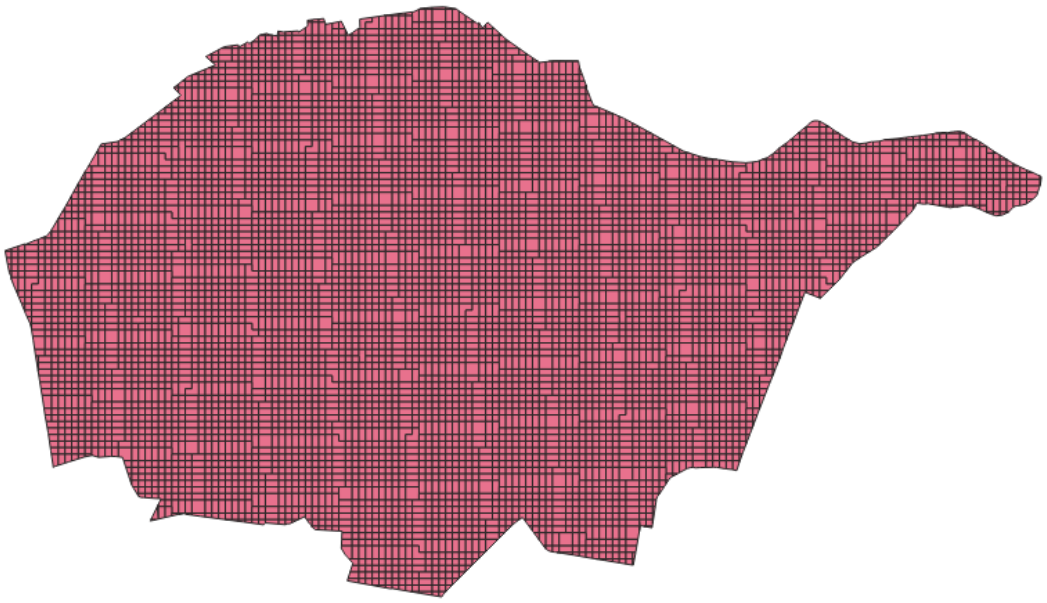
```
------------------------------------------------------------------

--Tworzenie rastrów z wektorów (rastrowanie)

------------------------------------------------------------------

--Poniższe przykłady pokazują rastrowanie wektoru.

--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 kwasniak.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 kwasniak.porto_parishes; --> drop table porto_parishes first

CREATE TABLE kwasniak.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';
```

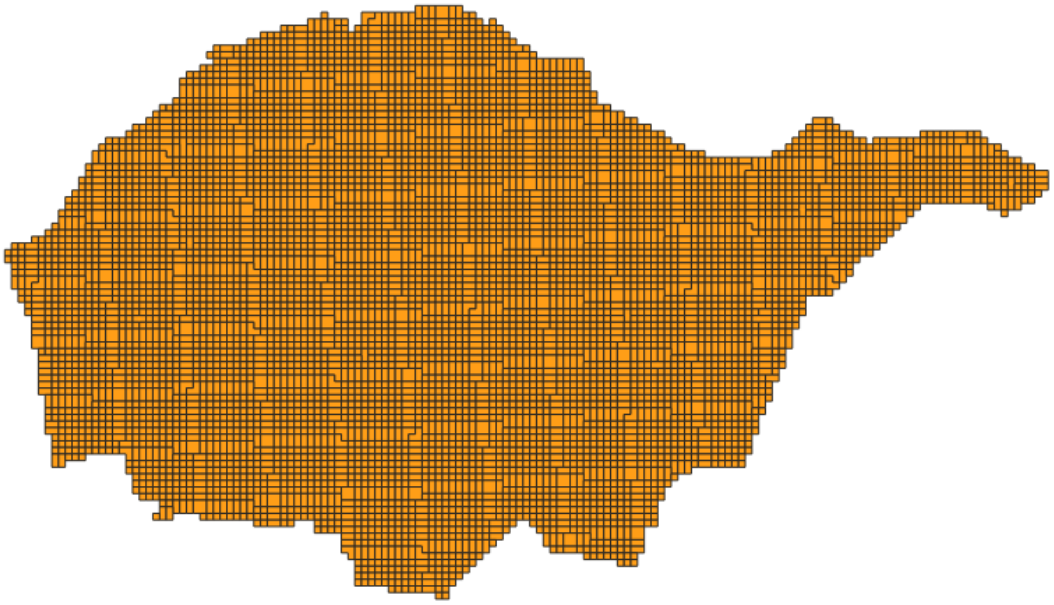
```
--Przykład 3 - ST_Tile
--Po uzyskaniu pojedynczego rastra można generować kafelki za pomocą funkcji ST_Tile.
DROP TABLE kwasniak.porto_parishes; --> drop table porto_parishes first
CREATE TABLE kwasniak.porto_parishes AS
WITH r AS (
SELECT rast FROM rasters.dem
LIMIT 1 )
SELECT st_tile(st_union(ST_AsRaster(a.geom,r.rast,'8BUI',a.id,-
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
--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 kwasniak.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 kwasniak.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);
```

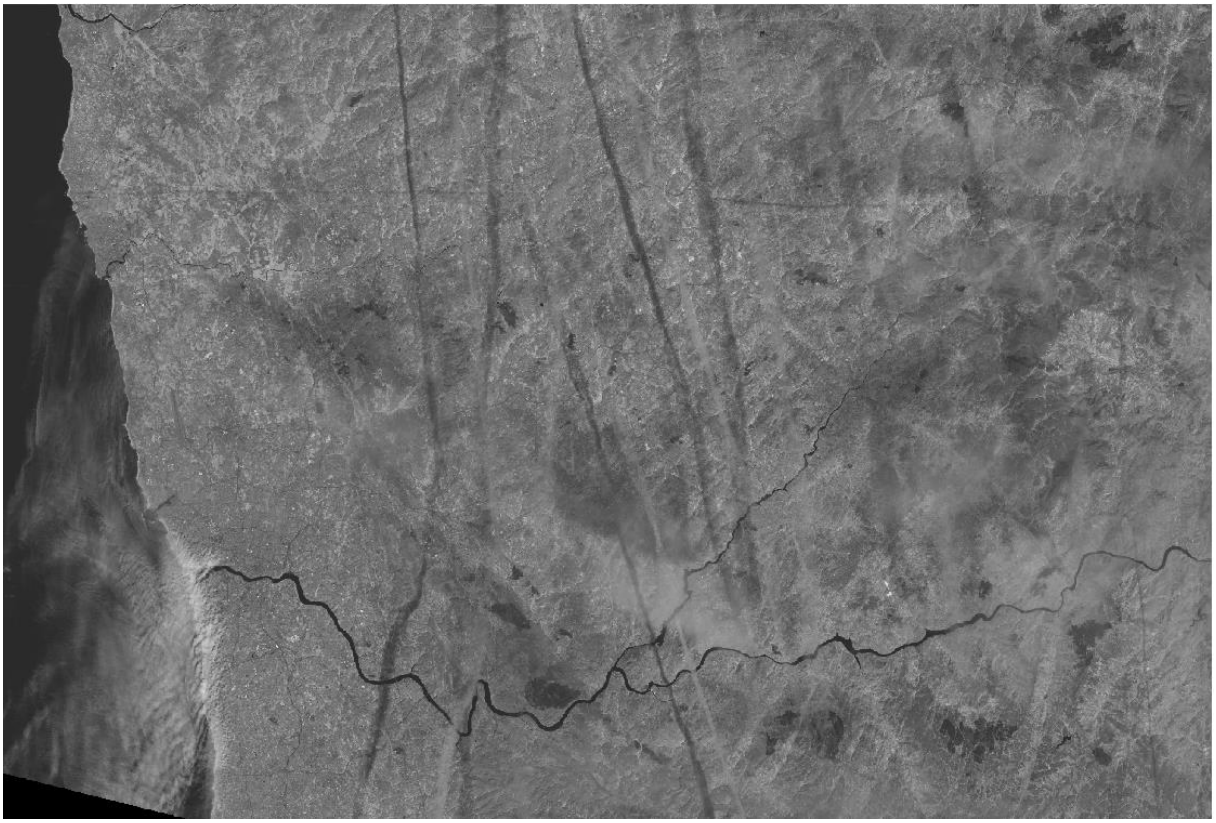


--Analiza rastrów

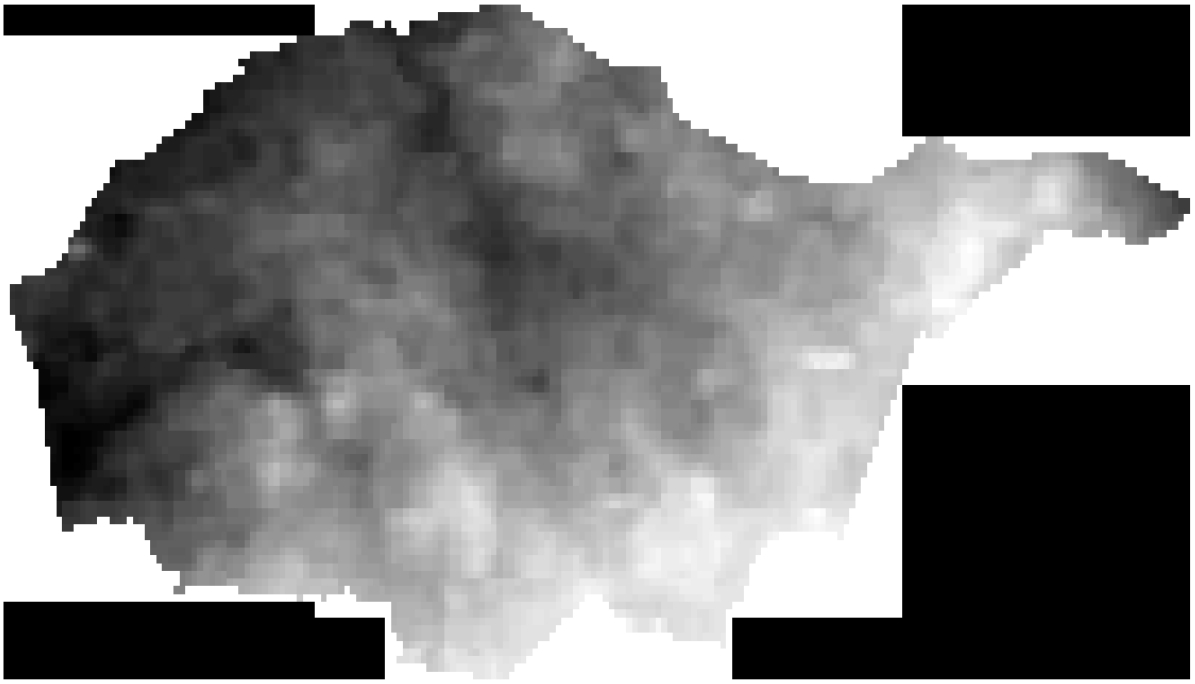
--Przykład 1 - ST_Band

--Funkcja ST_Band służy do wyodrębniania pasm z rastra

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

-- 4 - ST_Reclass

--Aby zreklasyfikować raster należy użyć funkcji ST_Reclass.

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

--Przykład 6 - ST_SummaryStats oraz Union

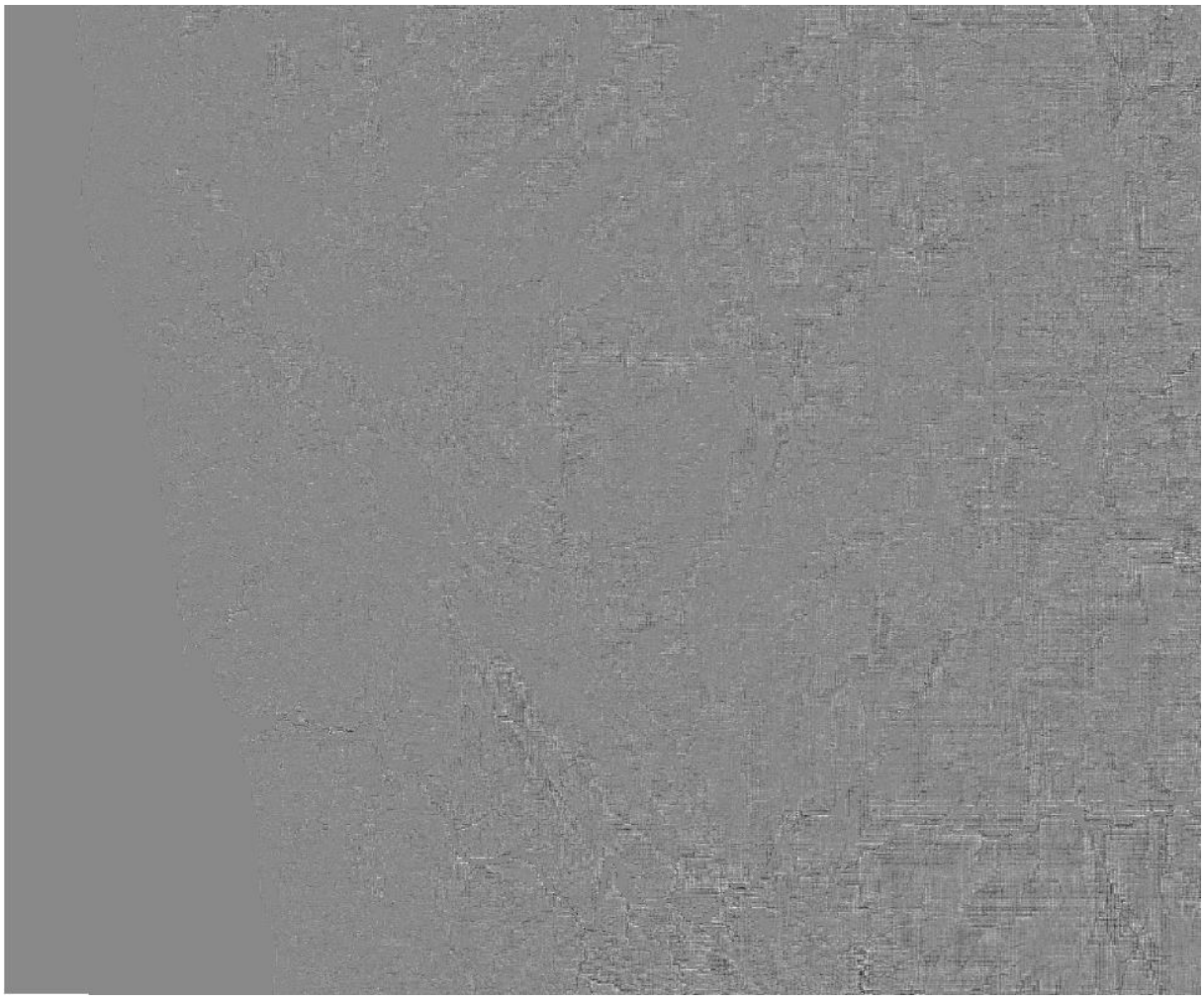
--Przy użyciu UNION można wygenerować jedną statystykę wybranego rastra.

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

--Przykład 7 - ST_SummaryStats z lepszą kontrolą złożonego typu danych

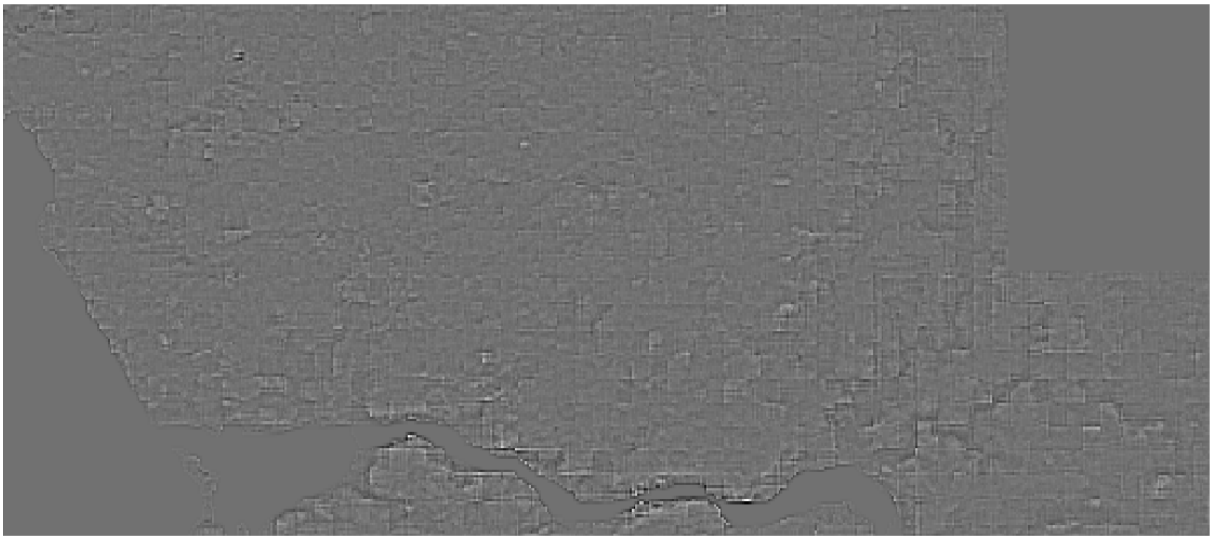
```
WITH t AS (
SELECT st_summarystats(ST_Union(a.rast)) AS stats
FROM kwasniak.paranhos_dem AS a
)
SELECT (stats).min,(stats).max,(stats).mean FROM t;

--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;

-------------------------------
--Topographic Position Index (TPI)
```

```
------------------------------

--Przykład 10 - ST_TPI

--jak obliczyć TPI przy użyciu tabeli rasters.dem jako danych wejściowych. Tabela nazywa się

--TPI30 ponieważ ma rozdzielczość 30 metrów i TPI używa tylko jednej komórki sąsiedztwa do obliczeń.

--Tabela wyjściowa z wynikiem zapytania zostanie stworzona w schemacie schema_name, jest więc

--możliwa jej wizualizacja w QGIS.

create table kwasniak.tpi30 as

select ST_TPI(a.rast,1) as rast

from rasters.dem a;


drop table kwasniak.tpi30;

--Poniższa kwerenda utworzy indeks przestrzenny:

CREATE INDEX idx_tpi30_rast_gist ON kwasniak.tpi30

USING gist (ST_ConvexHull(rast));


--Dodanie constraintów:

SELECT AddRasterConstraints('kwasniak'::name,

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



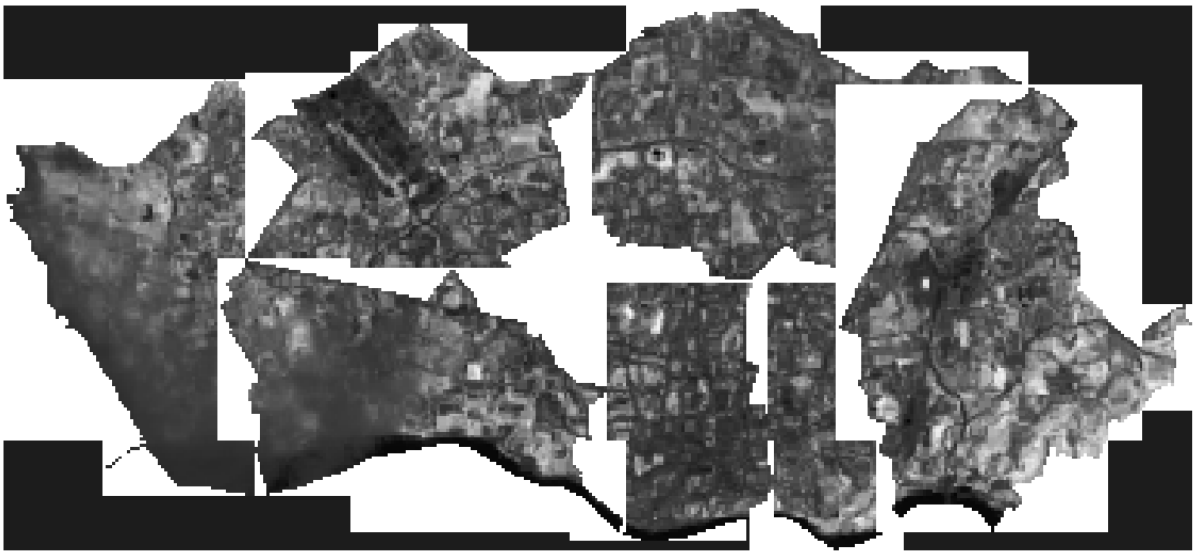
--Problem do samodzielnego rozwiązania

```
create table kwasniak.tpi30_ as
select ST_TPI(a.rast,1) as rast, b.municipality
from rasters.dem a, vectors.porto_parishes AS b
WHERE ST_Intersects(a.rast, b.geom) AND b.municipality ilike 'porto';
```



```
--------------------------
--Algebra map
--------------------------
--Istnieją dwa sposoby korzystania z algebry map w PostGIS. Jednym z nich jest użycie wyrażenia, a
--drugim użycie funkcji zwrotnej. Poniższe przykłady pokazują jak stosując obie techniki utworzyć
--wartości NDVI na podstawie obrazu Landsat8.
--Wzór na NDVI:
--NDVI=(NIR-Red)/(NIR+Red)
--Przykład 1 - Wyrażenie Algebry Map
CREATE TABLE kwasniak.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 kwasniak.porto_ndvi
USING gist (ST_ConvexHull(rast));
--Dodanie constraintów:
SELECT AddRasterConstraints('kwasniak'::name,
'porto_ndvi'::name,'rast'::name);
```



```
--Przykład 2 – Funkcja zwrotna
--W pierwszym kroku należy utworzyć funkcję, które będzie wywołana później:
create or replace function kwasniak.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 kwasniak.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],
'kwasniak.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 kwasniak.porto_ndvi2
USING gist (ST_ConvexHull(rast));
--Dodanie constraintów:
SELECT AddRasterConstraints('kwasniak'::name,
'porto_ndvi2'::name,'rast'::name);
```

--Eksport danych

--Przykład 1 - ST_AsTiff

--Funkcja ST_AsTiff tworzy dane wyjściowe jako binarną reprezentację pliku tiff, może to być przydatne

--na stronach internetowych, skryptach itp., w których programista może kontrolować, co zrobić z

--plikiem binarnym, na przykład zapisać go na dysku lub po prostu wyświetlić.

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

--Przykład 2 - ST_AsGDALRaster

--Podobnie do funkcji ST_AsTiff, ST_AsGDALRaster nie zapisuje danych wyjściowych bezpośrednio na

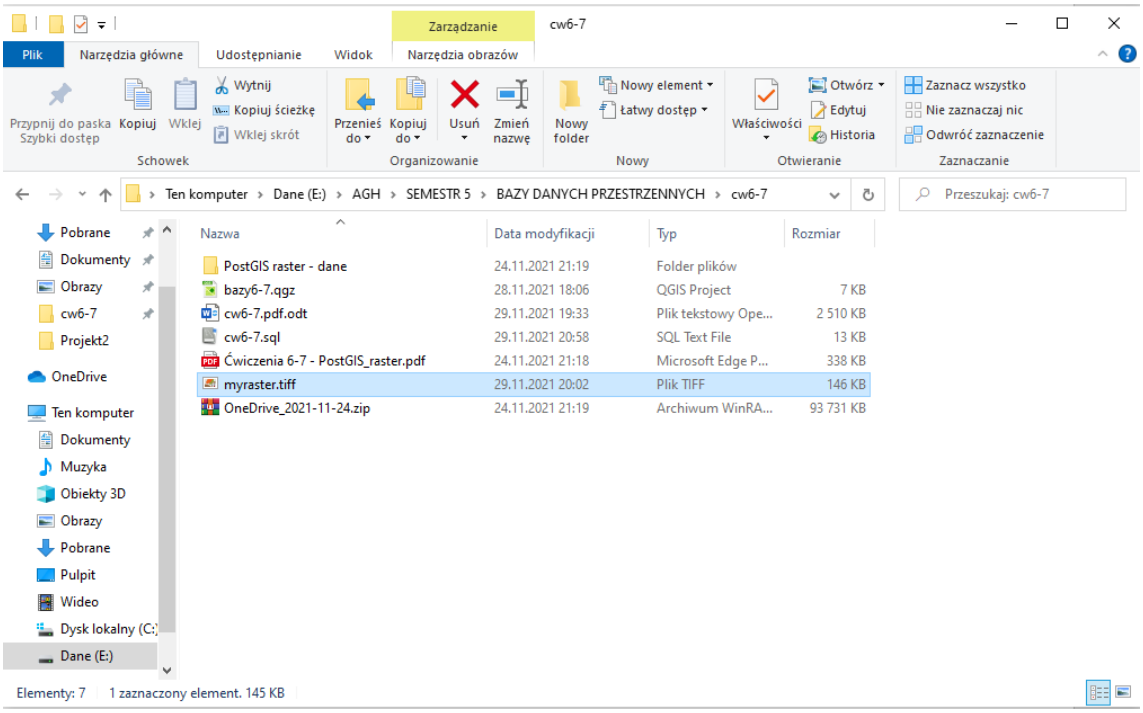
--dysku, natomiast dane wyjściowe są reprezentacją binarną dowolnego formatu GDAL.

```
SELECT ST_AsGDALRaster(ST_Union(rast), 'GTiff', ARRAY['COMPRESS=DEFLATE',
'PREDICTOR=2', 'PZLEVEL=9'])
FROM kwasniak.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 kwasniak.porto_ndvi;
----------------------------------------------
SELECT lo_export(loid, 'E:\AGH\SEMESTR 5\BAZY DANYCH PRZESTRZENNYCH\cw6-7\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.
```


Zarządzanie
cw6-7
Plik
Narzędzia główne
Udostępnianie
Widok
Narzędzia obrazów
Przypnij do paska Szybki dostęp
Kopiuj
Wklej
Wytnij
Kopiuj ścieżkę
Wklej skrót
Schowek
Przenieś do
Kopiuj do
Usuń
Zmień nazwę
Organizowanie
Nowy folder
Nowy element
Łatwy dostęp
Nowy
Właściwości
Otwórz
Edytuj
Historia
Otwieranie
Zaznacz wszystko
Nie zaznaczaj nic
Odwróć zaznaczenie
Zaznaczanie
Ten komputer › Dane (E:) › AGH › SEMESTR 5 › BAZY DANYCH PRZESTRZENNYCH › cw6-7
Przeszukaj: cw6-7
Pobrane
Dokumenty
Obrazy
cw6-7
Projekt2
OneDrive
Ten komputer
Dokumenty
Muzyka
Obiekty 3D
Obrazy
Pobrane
Pulpit
Wideo
Dysk lokalny (C:)
Dane (E:)
Nazwa
Data modyfikacji
Typ
Rozmiar
PostGIS raster - dane 24.11.2021 21:19 Folder plików
bazy6-7.qgz 28.11.2021 18:06 QGIS Project 7 KB
cw6-7.pdf.odt 29.11.2021 19:33 Plik tekstowy Ope... 2 510 KB
cw6-7.sql 29.11.2021 20:58 SQL Text File 13 KB
Ćwiczenia 6-7 - PostGIS_raster.pdf 24.11.2021 21:18 Microsoft Edge P... 338 KB
myraster.tiff 29.11.2021 20:02 Plik TIFF 146 KB
OneDrive_2021-11-24.zip 24.11.2021 21:19 Archiwum WinRA... 93 731 KB
Elementy: 7
1 zaznaczony element. 145 KB