Przykład 1

```
CREATE TABLE szymon_berdzik.porto_parishes AS
WITH r AS (
SELECT rast FROM rasters.dem
LIMIT 1
)

SELECT ST_AsRaster( geom: a.geom, scalex: r.rast, scaley: '8BUI', gridx: a.id, gridy: -32767) AS rast
FROM vectors.porto_parishes AS a, r
WHERE a.municipality ilike 'porto';
```

```
_parishes
[2022-11-28 08:20:17] Position: 98
postgis_raster.public> CREATE TABLE szymon_berdzik.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'
[2022-11-28 08:20:28] 7 rows affected in 74 ms
```

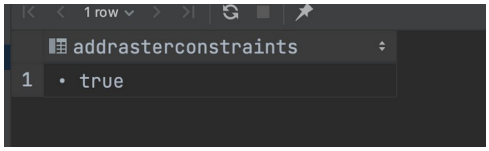
Przykład 1

```
postgis_raster.public> CREATE TABLE szymon_berdzik.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'
[2022-11-28 08:23:45] 25 rows affected in 22 ms
```

```
postgis_raster.public> alter table szymon_berdzik.intersects
                       add column rid SERIAL PRIMARY KEY
[2022-11-28 08:25:16] completed in 16 ms
```

```
postgis_raster.public> CREATE INDEX idx_intersects_rast_gist ON szymon_berdzik.intersects
                       USING gist (ST_ConvexHull(rast))
[2022-11-28 08:25:20] completed in 5 ms
```

```
SELECT AddRasterConstraints( rasttable: 'szymon_berdzik'::name,
 stcolumn: 'intersects'::name, srid: 'rast'::name);
```



Przykład 2 – ST Clip

```
[2022-11-28 08:25:24] 1 row retrieved starting from 1 in 247 ms (execution: 46 ms, fetching: 201 ms)
postgis_raster.public> CREATE TABLE szymon_berdzik.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'
[2022-11-28 08:27:43] 25 rows affected in 36 ms
```

Przykład 3 – ST_Union

```
postgis_raster.public> CREATE TABLE szymon_berdzik.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)
[2022-11-28 08:29:08] 1 row affected in 53 ms
```

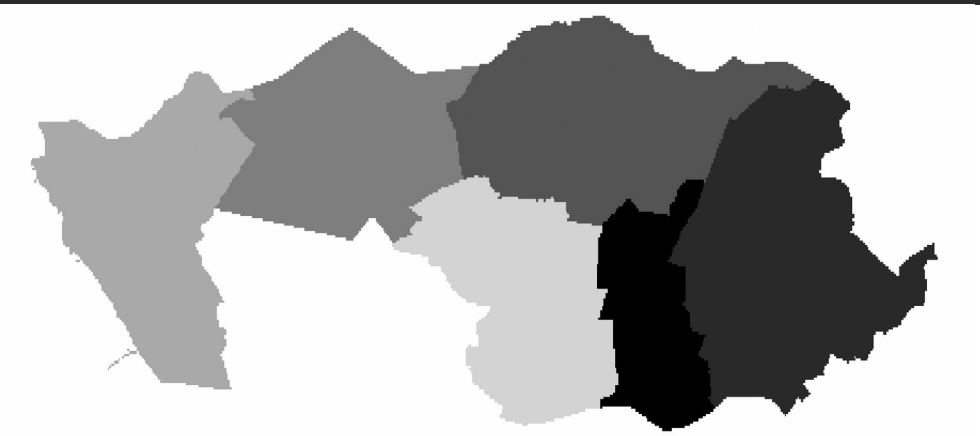
Tworzenie rastrów z wektorów

Przykład 1 – ST_AsRaster

```
postgis_raster.public> CREATE TABLE szymon_berdzik.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'
[2022-11-28 08:30:49] 7 rows affected in 20 ms
```

Przykład 2 – ST_Union

```
postgis_raster.public> DROP TABLE szymon_berdzik.porto_parishes
[2022-11-28 08:36:31] completed in 4 ms
postgis_raster.public> CREATE TABLE szymon_berdzik.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'
[2022-11-28 08:36:31] 1 row affected in 36 ms
```



Przykład 3 – ST_Tile – taki sam resultat jak wyzej

```
postgis_raster.public> DROP TABLE szymon_berdzik.porto_parishes
[2022-11-28 08:38:02] completed in 10 ms
postgis_raster.public> CREATE TABLE szymon_berdzik.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'
[2022-11-28 08:38:02] 8 rows affected in 38 ms
```

Konwertowanie rastrów na wektory

Przykład 1

```
postgis_raster.public> create table szymon_berdzik.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)
[2022-11-28 08:40:13] 6,629 rows affected in 1 s 670 ms
```

Przykład 2

```
postgis_raster.public> CREATE TABLE szymon_berdzik.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)
[2022-11-28 08:41:01] 6,422 rows affected in 61 ms
```

Analiza Rastrow

Przykład 1

```
postgis_raster.public> CREATE TABLE szymon_berdzik.landsat_nir AS
                       SELECT rid, ST_Band(rast,4) AS rast
                       FROM rasters.landsat8
[2022-11-28 08:41:56] 384 rows affected in 337 ms
```

Przykład 2

```
postgis_raster.public> CREATE TABLE szymon_berdzik.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)
[2022-11-28 08:42:46] 4 rows affected in 15 ms
```

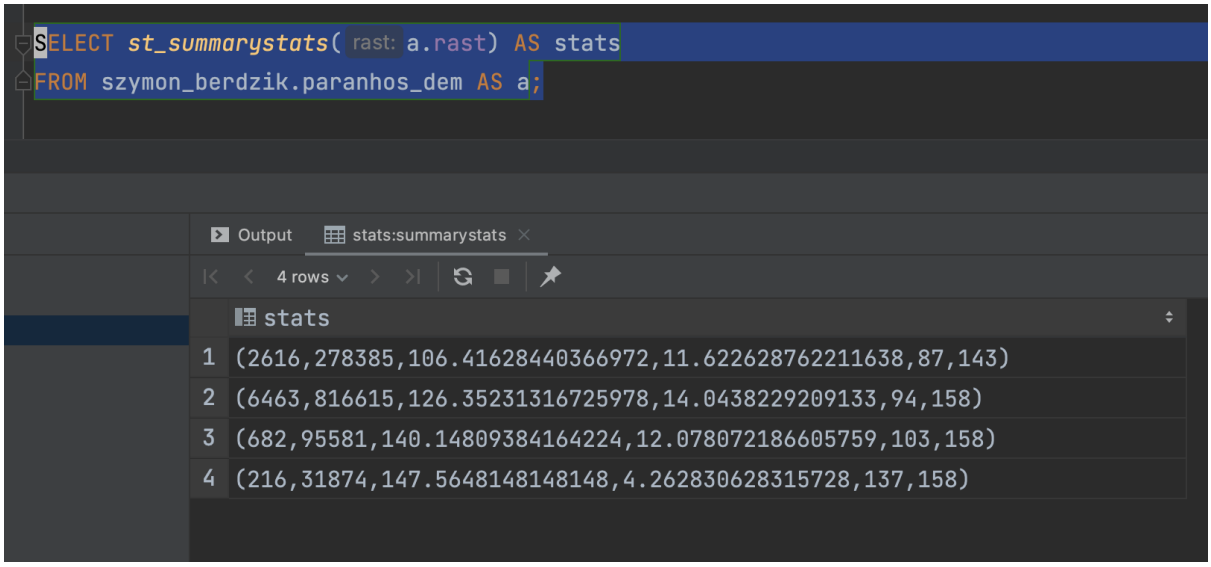
Przykład 3

```
postgis_raster.public> CREATE TABLE szymon_berdzik.paranhos_slope AS
                         SELECT a.rid,ST_Slope(a.rast,1,'32BF','PERCENTAGE') as rast
                         FROM szymon_berdzik.paranhos_dem AS a
[2022-11-28 08:43:31] 4 rows affected in 77 ms
```

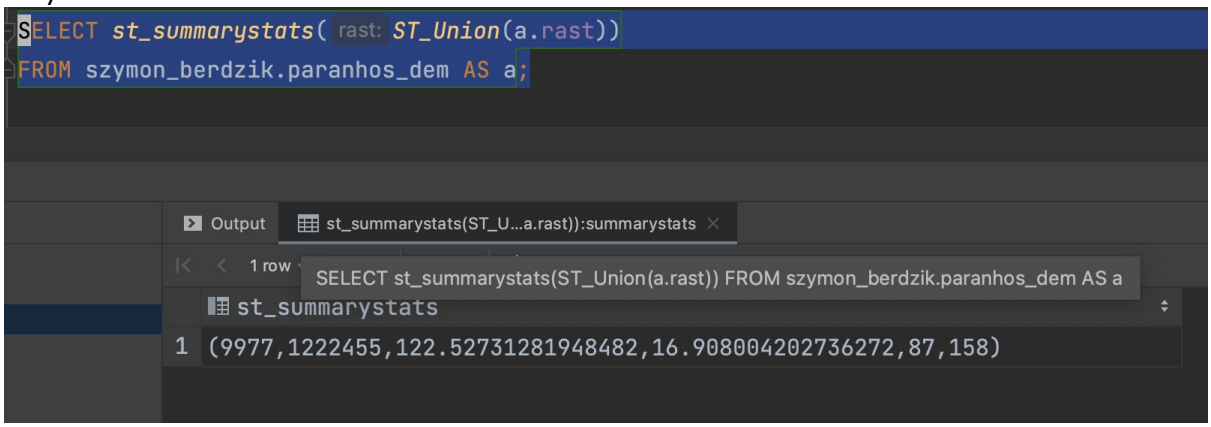
Przykład 4

```
postgis_raster.public> CREATE TABLE szymon_berdzik.paranhos_slope_reclass AS
                       SELECT a.rid,ST_Reclass(a.rast,1,']0-15]:1, (15-30]:2, (30-9999:3',
                       '32BF',0)
                       FROM szymon_berdzik.paranhos_slope AS a
[2022-11-28 08:44:10] 4 rows affected in 10 ms
```

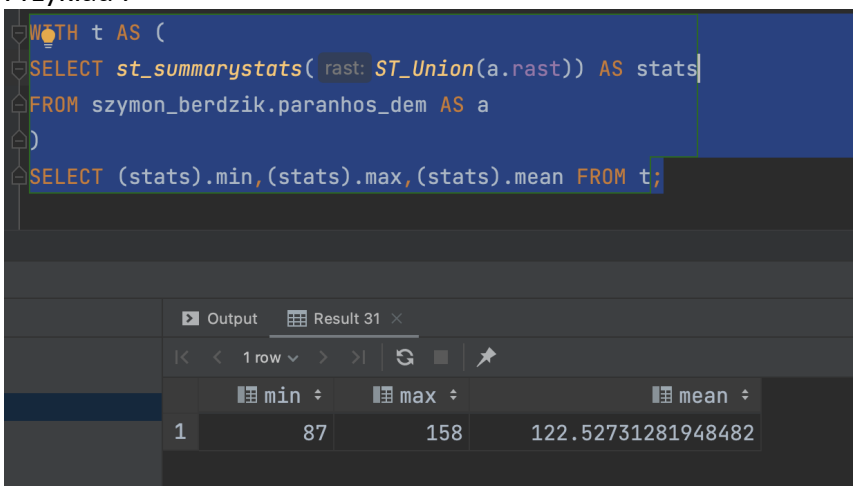
Przykład 5



Przykład 6



Przykład 7



Przykład 8

```
WITH t AS (
SELECT b.parish AS parish, st_summarystats( rast: ST_Union(ST_Clip( rast: a.rast,
 geom: b.geom, crop: true))) AS stats
 FROM rasters.dem AS a, vectors.porto_parishes AS b
 WHERE b.municipality ilike 'porto' and ST_Intersects( geog1: b.geom, geog2: a.rast)
group by b.parish
)
SELECT parish,(stats).min,(stats).max,(stats).mean FROM t;
```

Output | Result 32

7 rows | CSV

	parish	min	max	mean
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…	1	157	95.00277741039545
7	União das freguesias de Lordelo do Ouro e Massarelos	-1	117	49.50051440329218

Przykład 9

```
SELECT b.name,st_value( rast: a.rast, pt: (ST_Dump(b.geom)).geom)
FROM
rasters.dem a, vectors.places AS b
WHERE ST_Intersects( geog1: a.rast, geog2: b.geom)
ORDER BY b.name;
```

Output | Result 33

33 rows

	name	st_value
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

Topographic Position Index (TPI)

Przykład 10

```
postgis_raster.public> create table szymon_berdzik.tpi30 as
                       select ST_TPI(a.rast,1) as rast
                       from rasters.dem a
[2022-11-28 08:49:35] 589 rows affected in 13 s 566 ms
postgis_raster.public> CREATE INDEX idx_tpi30_rast_gist ON szymon_berdzik.tpi30
                       USING gist (ST_ConvexHull(rast))
[2022-11-28 08:49:35] completed in 3 ms
postgis_raster.public> SELECT AddRasterConstraints('szymon_berdzik'::name,
                       'tpi30'::name,'rast'::name)
```

```
postgis_raster.public> create table szymon_berdzik.tpi30_porto as
                       select ST_TPI(a.rast,1) as rast
                       from rasters.dem a, vectors.porto_parishes as p
                       where st_intersects(a.rast, p.geom) and  p.municipality like 'porto'
[2022-11-28 08:53:00] completed in 9 ms
```

Algebra map

```
postgis_raster.public> CREATE TABLE szymon_berdzik.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
[2022-11-28 08:55:27] 23 rows affected in 122 ms
postgis_raster.public> CREATE INDEX idx_porto_ndvi_rast_gist ON szymon_berdzik.porto_ndvi
                       USING gist (ST_ConvexHull(rast))
[2022-11-28 08:56:00] completed in 8 ms
postgis_raster.public> SELECT AddRasterConstraints('szymon_berdzik'::name,
                       'porto_ndvi'::name,'rast'::name)
Adding SRID constraint
Adding scale-X constraint
Adding scale-Y constraint
Adding blocksize-X constraint
Adding blocksize-Y constraint
Adding alignment constraint
Adding number of bands constraint
Adding pixel type constraint
Adding nodata value constraint
Adding out-of-database constraint
Adding maximum extent constraint
[2022-11-28 08:56:14] 1 row retrieved starting from 1 in 33 ms (execution: 24 ms, fetching: 9 ms)
```

Przykład 2

```
postgis_raster.public> create or replace function szymon_berdzik.ndvi(
                       value double precision [] [] [],
                       pos integer [][],
                       VARIADIC userargs text []
                       )
                       RETURNS double precision AS
                       $$
                       BEGIN
                       --RAISE NOTICE 'Pixel Value: %', value [1][1][1];-->For debug

                       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
[2022-11-28 08:58:03] completed in 3 ms
```

```
postgis_raster.public> CREATE TABLE szymon_berdzik.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],
                       'szymon_berdzik.ndvi(double precision[],
                       integer[],text[])'::regprocedure, --> This is the function!
                       '32BF'::text
                       ) AS rast
                       FROM r
[2022-11-28 08:58:03] 23 rows affected in 99 ms
postgis_raster.public> CREATE INDEX idx_porto_ndvi2_rast_gist ON szymon_berdzik.porto_ndvi2
                       USING gist (ST_ConvexHull(rast))
[2022-11-28 08:58:03] completed in 2 ms
```

Eksport danych

Przykład 1 – binarna reprezentacja pliku

```
SELECT ST_AsTiff( rast: ST_Union(rast))
FROM szymon_berdzik.porto_ndvi;
```

Output | ST_AsTiff(ST_Union(rast)):bytea

1 row

	st_astiff
1	204.8kB of 275.54kB 0x49492A00080000001000000103000

Przykład 2 – dowoly format gd

```
SELECT ST_AsGDALRaster( rast: ST_Union(rast), format: 'GTiff', options: ARRAY['COMPRESS=DEFLATE',
'PREDICTOR=2', 'PZLEVEL=9'])
FROM szymon_berdzik.porto_ndvi;
```

Output | ST_AsGDALRaster(ST_U..., 'PZLEVEL=9']):bytea

1 row

	st_asgdalraster
1	384x179 TIFF image 148.84 kB

Przykład 3

```
CREATE TABLE tmp_out AS
SELECT lo_from_bytea(0,
 ST_AsGDALRaster( rast: ST_Union(rast),  format: 'GTiff',  options: ARRAY['COMPRESS=DEFLATE',
'PREDICTOR=2', 'PZLEVEL=9'])
 ) AS loid
FROM szymon_berdzik.porto_ndvi;

SELECT lo_export(loid, '/Users/szymonberdzik/studia/bazy/bazymyraster.tiff')  FROM tmp_out;
```

Output | lo_export(loid, '/Us...raster.tiff'):integer

1 row

	lo_export
1	1

Przykład 4

```
➜  bazy git:(master) ✗ gdal_translate -co COMPRESS=DEFLATE -co PREDICTOR=2 -co ZLEVEL=9 PG:"host
=localhost port=5432 dbname=postgis_raster user=postgres  schema=szymon_berdzik table=porto_ndvi
 mode=2" porto_ndvi.tiff
Input file size is 384, 179
0...10...20...30...40...50...60...70...80...90...100 - done.
```