

ST_intersects

```
1 CREATE TABLE schema_name.intersects AS
2 SELECT a.rast, b.municipality
3 FROM rasters.dem AS a, vectors.porto_parishes AS b
4 WHERE ST_Intersects(geog1 a.rast, geog2 b.geom) AND b.municipality ilike 'porto';
5
6 ✓ SELECT
7     st_width(rast) as width,
8     st_height(rast) as height,
9     st_srid(geog rast) as srid,
10    st_numbands(rast) as bands
11 FROM schema_name.intersects
```

Output Result 8 x

	width ▾	height ▾	srid ▾	bands ▾
1	100	100	3763	1
2	100	100	3763	1
3	100	100	3763	1
4	100	100	3763	1
5	100	100	3763	1
6	100	100	3763	1
7	100	100	3763	1
8	100	100	3763	1
9	100	100	3763	1
10	100	100	3763	1
11	100	100	3763	1
12	100	100	3763	1

25 rows ▾

```
16 ✓ CREATE INDEX idx_intersects_rast_gist ON schema_name.intersects USING GIST (st_convexhull(rast));
17
18 ✓ select addrasterconstraints( rasttable 'schema_name'::name, rastcolumn 'intersects'::name, srid 'rast'::name);
19
20 ✓ SELECT * FROM raster_columns WHERE r_table_name = 'intersects';
21
```

Output addrasterconstraints...rast::name):boolean

sprawdzenie czy ogra...czenia zostały dodane x

	r_table_catalog ▾	r_table_schema ▾	r_table_name ▾	r_raster_column ▾	srid ▾	scale_x
1	postgis_raster	schema_name	intersects	rast	3763	23.3527

ST_clip

```
CREATE TABLE schema_name.clip AS
SELECT ST_Clip(rast a.rast, geom b.geom, crop true) AS rast, b.municipality
FROM rasters.dem AS a, vectors.porto_parishes AS b
WHERE ST_Intersects(geog1 a.rast, geog2 b.geom) AND b.municipality like 'PORTO';

SELECT
  st_width(rast) as width,
  st_height(rast) as height,
  st_srid(geog rast) as srid,
  st_numbands(rast) as bands
FROM schema_name.clip
```

Output Result 6 x sprawdzenie czy ogra...czenia zostały dodane



	width ▾	height ▾	srid ▾	bands ▾
1	66	93	3763	1
2	48	11	3763	1
3	17	30	3763	1
4	70	54	3763	1
5	100	87	3763	1
6	31	83	3763	1
7	15	32	3763	1

25 rows ▾

ST_union

```
CREATE TABLE schema_name.union AS
SELECT ST_Union(ST_Clip(rast a.rast, geom b.geom, crop true))
FROM rasters.dem AS a, vectors.porto_parishes AS b
WHERE b.municipality ilike 'porto' and ST_Intersects(geom1 b.geom, geog2 a.rast);

SELECT
  st_width(st_union) as width,
  st_height(st_union) as height,
  st_srid(geog st_union) as srid,
  st_numbands(st_union) as bands
FROM schema_name.union;
```

	width	height	srid	bands
1	498	172	3763	1

ST_asraster

```
50 ✓ CREATE TABLE schema_name.porto_parishes AS
51 WITH r AS (
52   SELECT rast FROM rasters.dem
53   LIMIT 1
54 )
55 SELECT ST_AsRaster(geom a.geom, scalex r.rast, scaley '8BUI', gridx a.id, gridy -32767) AS rast
56 FROM vectors.porto_parishes AS a, r
57 WHERE a.municipality ilike 'porto';
58
59 ✓ SELECT
60   st_width(rast) as width,
61   st_height(rast) as height,
62   st_srid(geog rast) as srid,
63   st_numbands(rast) as bands
64 FROM schema_name.porto_parishes;
```

	width	height	srid	bands
1	66	105	3763	1
2	149	142	3763	1
3	146	90	3763	1
4	125	133	3763	1
5	202	88	3763	1
6	125	104	3763	1
7	171	87	3763	1

7 rows

ST_union

```
67 ✓ DROP TABLE schema_name.porto_parishes; → drop table porto_parishes first
68 ✓ CREATE TABLE schema_name.porto_parishes AS
69 WITH r AS (
70     SELECT rast FROM rasters.dem
71     LIMIT 1
72 )
73 SELECT st_union(ST_AsRaster( geom a.geom, scalex r.rast, scaley '8BUI', gridx a.id, gridy -32767)) AS rast
74 FROM vectors.porto_parishes AS a, r
75 WHERE a.municipality ilike 'porto';
76
77 ✓ SELECT
78     st_width(rast) as width,
79     st_height(rast) as height,
80     st_srid(geog_rast) as srid,
81     st_numbands(rast) as bands
82 FROM schema_name.porto_parishes;
```

Output Result 13-2 Result 14-3 ×

width height srid bands

	width	height	srid	bands
1	499	173	3763	1

ST_tile

```
85 ✓ DROP TABLE schema_name.porto_parishes; → drop table porto_parishes first
86 ✓ CREATE TABLE schema_name.porto_parishes AS
87 WITH r AS (
88     SELECT rast FROM rasters.dem
89     LIMIT 1 )
90 SELECT st_tile( rast st_union(ST_AsRaster( geom a.geom, scalex r.rast, scaley '8BUI', gridx a.id,
91     gridy 32767)), width 128, height 128, padwithnodata true, nodataval -32767) AS rast
92 FROM vectors.porto_parishes AS a, r
93 WHERE a.municipality ilike 'porto';
94
95 ✓ SELECT
96     st_width(rast) as width,
97     st_height(rast) as height,
98     st_srid(geog_rast) as srid,
99     st_numbands(rast) as bands
100 FROM schema_name.porto_parishes;
```

Output Result 15-3 × Result 14-3

width height srid bands

	width	height	srid	bands
1	128	128	3763	1
2	128	128	3763	1
3	128	128	3763	1
4	128	128	3763	1
5	128	128	3763	1
6	128	128	3763	1
7	128	128	3763	1

8 rows

ST_intersection

```
105 create table schema_name.intersection as
106 SELECT
107     a.rid, (ST_Intersection(b.geom,a.rast)).geom, (ST_Intersection(b.geom,a.rast)
108     ).val
109 FROM rasters.landsat8 AS a, vectors.porto_parishes AS b
110 WHERE b.parish ilike 'paranhos' and ST_Intersects( geog1 b.geom, geog2 a.rast);
111
112 ✓ SELECT rid, st_astext(geom), val FROM schema_name.intersection;
113
```

Output Result 15-3 Result 23 x

	rid	st_astext	val
1	221	POLYGON((-39604.86528035818 168624.02902255123,-39633.58795484908 168624.0...	10648
2	221	POLYGON((-39574.553878279854 168624.02902255123,-39604.86528035818 168624.0...	12155
3	221	POLYGON((-39786.73369282809 168594.3232286338,-39794.43786661896 168594.32...	9248
4	221	POLYGON((-39756.42229074977 168594.3232286338,-39786.73369282809 168594.32...	10030
5	221	POLYGON((-39726.11088867145 168594.3232286338,-39756.42229074977 168594.32...	10347
6	221	POLYGON((-39695.79948659313 168594.3232286338,-39726.11088867145 168594.32...	10347
7	221	POLYGON((-39665.488084514815 168594.3232286338,-39695.79948659313 168594.32...	10347

ST_dumpaspolygons

```
115 CREATE TABLE schema_name.dumppolygons AS
116 SELECT
117     a.rid, (ST_DumpAsPolygons( rast ST_Clip( rast a.rast, geom b.geom))).geom, (ST_DumpAsPolygons( rast ST_Clip( rast a.rast, geom b.geom)).val
118 FROM rasters.landsat8 AS a, vectors.porto_parishes AS b
119 WHERE b.parish ilike 'paranhos' and ST_Intersects( geog1 b.geom, geog2 a.rast);
120
121 ✓ SELECT rid, st_astext(geom), val FROM schema_name.dumppolygons;
```

Output Result 26 x

	rid	st_astext	val
1	221	POLYGON((-39665.488084514815 168624.02902255123,-39665.488084514815 1685...	12761
2	221	POLYGON((-39635.17668243649 168624.02902255123,-39635.17668243649 168564...	14382
3	221	POLYGON((-39604.86528035817 168624.02902255123,-39604.86528035817 168564...	14090
4	221	POLYGON((-40089.84771361128 168594.3232286338,-40089.84771361128 168564...	10625
5	221	POLYGON((-39907.979301141364 168594.3232286338,-39907.979301141364 16853...	8601
6	221	POLYGON((-39817.0450949064 168594.3232286338,-39817.0450949064 16853...	10347
7	221	POLYGON((-39786.733692828086 168594.3232286338,-39786.733692828086 16853...	10347

ST_band

```
124 ✓ CREATE TABLE schema_name.landsat_nir AS
125 SELECT rid, ST_Band(rast rast, nband 4) AS rast
126 FROM rasters.landsat8;
127
128 ✓ SELECT
129     st_width(rast) as width,
130     st_height(rast) as height,
131     st_srid(geog rast) as srid,
132     st_numbands(rast) as bands
133 FROM schema_name.landsat_nir;
```

Output Result 27-2 x

	width ▾	height ▾	srid ▾	bands ▾
1	128	128	3763	1
2	128	128	3763	1
3	128	128	3763	1
4	128	128	3763	1
5	128	128	3763	1
6	128	128	3763	1
7	128	128	3763	1

384 rows ▾ ⋮

ST_clip

```
136 ✓ CREATE TABLE schema_name.paranhos_dem AS
137 SELECT a.rid, ST_Clip(rast a.rast, geom b.geom, crop true) as rast
138 FROM rasters.dem AS a, vectors.porto_parishes AS b
139 WHERE b.parish ilike 'paranhos' and ST_Intersects(geog1 b.geom, geog2 a.rast);
140
141 ✓ SELECT
142     st_width(rast) as width,
143     st_height(rast) as height,
144     st_srid(geog rast) as srid,
145     st_numbands(rast) as bands
146 FROM schema_name.paranhos_dem;
```

Output Result 28-2 x

	width ▾	height ▾	srid ▾	bands ▾
1	53	73	3763	1
2	49	32	3763	1
3	59	8	3763	1
4	100	79	3763	1

4 rows ▾ ⋮

ST_slope

```
149 ✓ CREATE TABLE schema_name.paranhos_slope AS
150   SELECT a.rid, ST_Slope( rast a.rast, nband 1, customextent '32BF', pixeltype 'PERCENTAGE') as rast
151   FROM schema_name.paranhos_dem AS a;
152
153 ✓ SELECT
154     st_width(rast) as width,
155     st_height(rast) as height,
156     st_srid( geog rast) as srid,
157     st_numbands(rast) as bands
158   FROM schema_name.paranhos_slope;
```

Output Result 29-2 x



	width ▾	÷	height ▾	÷	srid ▾	÷	bands ▾	÷
1	53		73		3763		1	
2	49		32		3763		1	
3	59		8		3763		1	
4	100		79		3763		1	

4 rows x 9

ST_reclass

```
CREATE TABLE schema_name.paranhos_slope_reclass AS
SELECT a.rid, ST_Reclass( rast a.rast, nband 1, reclassexpr ']'0-15]:1, (15-30]:2, (30-9999:3',
                        pixeltype '32BF', nodataval 0)
FROM schema_name.paranhos_slope AS a;
💡
SELECT * FROM schema_name.paranhos_slope_reclass;
```

Output postgis_raster.schem...aranhos_slope_reclass x



	rid ▾	÷	st_reclass ▾	÷
1	476		01000001006172BF3E4D5A374080318D6907CA3EC09A49D3957D46E4C033B2707F2F920441...	
2	478		01000001006172BF3E4D5A374080318D6907CA3EC02E3C8390DE87E2C0D7D06D6CAD850441...	
3	516		01000001006172BF3E4D5A374080318D6907CA3EC0187635E2BF88E3C0474F11FE054A0441...	
4	477		01000001006172BF3E4D5A374080318D6907CA3EC044951356C7ABE3C0DAE35DC008960441...	

ST_summarystats

```
-- ST_SummaryStats  
SELECT st_summarystats( rast a.rast) AS stats  
FROM schema_name.paranhos_dem AS a;
```

Output

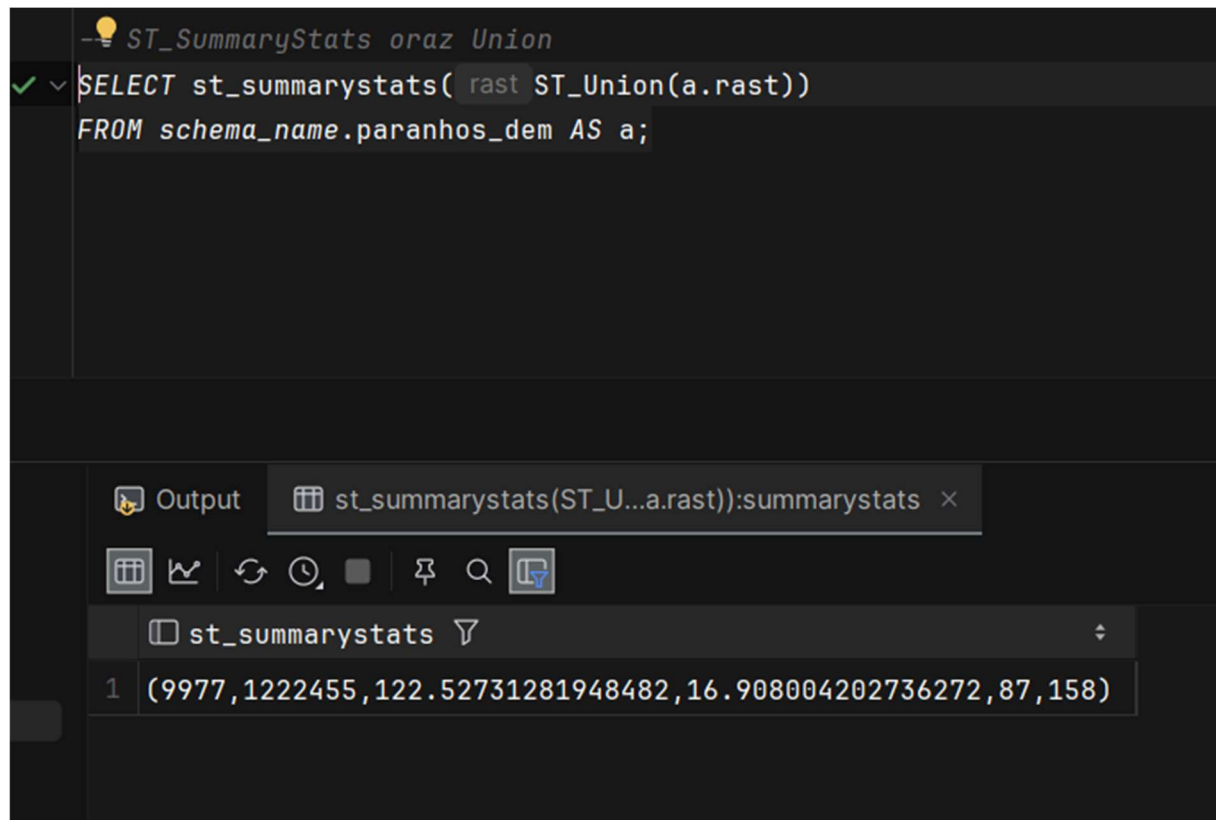
stats:summarystats x



stats

1	(2616,278385,106.41628440366972,11.622628762211638,87,143)
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)

ST_summarystats oraz union



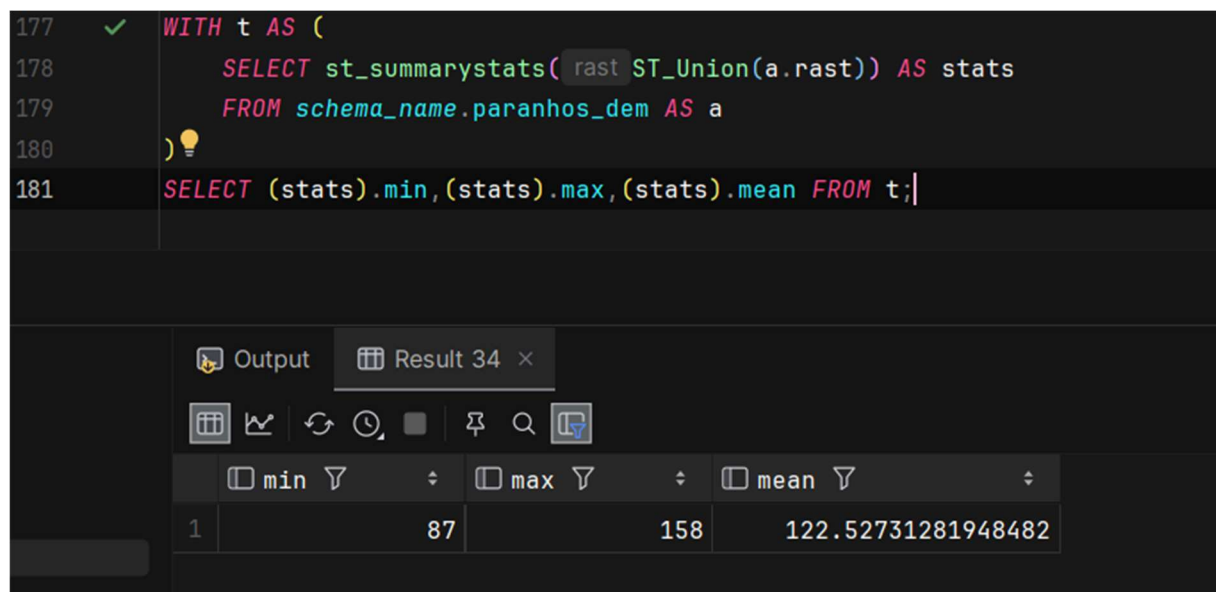
The screenshot shows a SQL IDE with a query editor and an output panel. The query in the editor is:

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

The output panel shows the result of the query, which is a single row of summary statistics:

st_summarystats
(9977,1222455,122.52731281948482,16.908004202736272,87,158)

ST_SummaryStats z lepszą kontrolą złożonego typu danych



The screenshot shows a SQL IDE with a query editor and an output panel. The query in the editor is:

```
177 WITH t AS (
178     SELECT st_summarystats( rast ST_Union(a.rast)) AS stats
179     FROM schema_name.paranhos_dem AS a
180 )
181 SELECT (stats).min,(stats).max,(stats).mean FROM t;
```

The output panel shows the result of the query, which is a single row of summary statistics:

min	max	mean
87	158	122.52731281948482

ST_SummaryStats w połączeniu z GROUP BY

```
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 35 x

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 e Vitória	1	157	95.00277741039545
7 União das freguesias de Lardelo do Douro e Massarelos	-1	117	49.50051640329218

ST_Value

```
194 SELECT b.name,st_value( rast a.rast, pt (ST_Dump(b.geom)).geom)  
195 FROM  
196     rasters.dem a, vectors.places AS b  
197 WHERE ST_Intersects( geog1 a.rast, geog2 b.geom)  
198 ORDER BY b.name;  
199
```

Output Result 36 x

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

ST_TPI

```
201 ✓ create table schema_name.tpi30 as
202 select ST_TPI( rast a.rast, nband 1) as rast
203 from rasters.dem a;
204
205 ✓ SELECT st_summarystats( rast tpi.rast)
206 FROM schema_name.tpi30 AS tpi;
```

Output st_summarystats(tpi.rast):summarystats x



st_summarystats ▾

1	(10000,0,0,0,0,0)
2	(10000,0,0,0,0.4727115928343625,-5.625,5)
3	(10000,0,0,0,0.7806607778030107,-5.25,5.125)
4	(10000,0,0,0,0.9911814289018942,-7.25,7)
5	(10000,0,0,1,512368795962151,-8.25,9.625)
6	(10000,0,0,1,4073447945037583,-9.75,12.625)

```
208 CREATE INDEX idx_tpi30_rast_gist ON schema_name.tpi30
209 USING gist (ST_ConvexHull(rast));|
210
211 ✓ SELECT AddRasterConstraints( rasttable 'schema_name'::name,
212                                rastcolumn 'tpi30'::name, srid 'rast'::name);
```

Output AddRasterConstraints...'rast'::name):boolean x



addrasterconstraints ▾

1	• true
---	--------

```

214 --maly obszar
215 ✓ create table schema_name.tpi30_porto as
216 select ST_TPI( rast a.rast, nband 1) as rast
217 from rasters.dem a, vectors.porto_parishes b
218 where b.municipality ilike 'porto' and ST_Intersects( geog1 a.rast, geog2 b.geom);
219
220 ✓ SELECT st_summarystats( rast tpi.rast)
221 FROM schema_name.tpi30_porto AS tpi;

```

Output st_summarystats(tpi.rast):summarystats ×

st_summarystats

1	(10000,0,0,1.950745049974495,-17.875,22.125)
2	(10000,0,0,1.1329193484092333,-6.25,8.375)
3	(10000,0,0,1.3757214016653214,-6.75,10.25)
4	(10000,0,0,1.2914212132375713,-7,9)
5	(10000,0,0,1.59291301551591,-9.875,11.625)
6	(10000,0,0,1.950745049974495,-17.875,22.125)
7	(10000,0,0,1.1329193484092333,-6.25,8.375)

25 rows

Wyrażenie Algebra Map

```
✓ CREATE TABLE schema_name.porto_ndvi AS
WITH r AS (
  SELECT a.rid, ST_Clip(rast a.rast, geom b.geom, crop true) AS rast
  FROM rasters.landsat8 AS a, vectors.porto_parishes AS b
  WHERE b.municipality ilike 'porto' and ST_Intersects(geom1 b.geom, geom2 a.rast)
)
SELECT
  r.rid, ST_MapAlgebra(
    rastbandargset r.rast, callbackfunc 1,
    pixelftype r.rast, extenttype 4,
    customextent '([rast2.val] - [rast1.val]) / ([rast2.val] +
[rast1.val])::float', distancex '32BF'
  ) AS rast
FROM r;

✓ SELECT st_summarystats(rast p.rast)
FROM schema_name.porto_ndvi AS p;
```

Output st_summarystats(p.rast):summarystats x

st_summarystats

1	(0,,,,)
2	(1245,191.9515317317564,0.15417793713394087,0.11066626399581464,-0.05336048826575279,0.5319941639900208)
3	(2196,280.6113417702727,0.1277829425183391,0.06599777954104259,0.0012179126497358084,0.5395634770393372)
4	(1150,217.55006091190444,0.1891739660103517,0.1454306238620828,-0.05492142215371132,0.5156594514846802)
5	(270,80.92967846244574,0.2997395498609101,0.0981692238693096,0.08118022233247757,0.510151743888855)
6	(7516,1388.5440665324859,0.18474508602082038,0.09954936911935036,-0.05026611313223839,0.5750077021110776)
7	(139,24.92294780910015,0.17930178279928166,0.06421807010904389,0.065883107483387,0.368705

23 rows

Funkcja zwrotna

```
✓ CREATE OR REPLACE FUNCTION schema_name.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;
```

```
    VARIADIC userargs text[]  
    )  
    RETURNS double precision AS  
$$  
BEGIN  
    -- RAISE NOTICE 'Pixel Value: %', value[1][1][1]; -- For debug purposes  
    RETURN (value[2][1][1] - value[1][1][1]) / (value[2][1][1] + value[1][1][1]); --  
END;  
$$  
  
LANGUAGE 'plpgsql' IMMUTABLE COST 1000  
[2024-12-09 01:47:44] completed in 4 ms
```

Funkcje TPI

```
CREATE TABLE schema_name.porto_ndvi2 AS  
with r as (  
    select a.rid, ST_Clip(rast a.rast, geom b.geom, crop true) as rast  
    from rasters.landsat8 a, vectors.porto_parishes b  
    where b.municipality ilike 'porto' and ST_Intersects(geom1 a.rast, geom2 b.geom)  
    )  
select r.rid, ST_MapAlgebra(rastbandargset r.rast, callbackfunc ARRAY [1,4], pixeltypes 'schema_name.ndvi(double precision[][][], integer[][], text[])::r  
from r;  
  
SELECT * FROM schema_name.porto_ndvi2;
```

Output postgis_raster.schema_name.porto_ndvi2 x

rid	rast
1	245 01000001003849EE08B84F3E404F2001E9AEB43DC0086F60B09B56E3C01F1195122C3204...
2	270 01000001003849EE08B84F3E404F2001E9AEB43DC075735DC2074FE3C0E4374673AE1104...
3	246 01000001003849EE08B84F3E404F2001E9AEB43DC03EF15EB9D152E3C075C249A2725504...
4	270 01000001003849EE08B84F3E404F2001E9AEB43DC03E4E2033AFB3E2C0E4374673AE1104...
5	247 01000001003849EE08B84F3E404F2001E9AEB43DC0AB0CA038D66DE1C079E13D89743B04...
6	246 01000001003849EE08B84F3E404F2001E9AEB43DC075B133BEF0E4E2C00224CA124B7E04...
7	221 01000001003849EE08B84F3E404F2001E9AEB43DC0C069108BB315E5C0B6C41B00DC9004...

23 rows