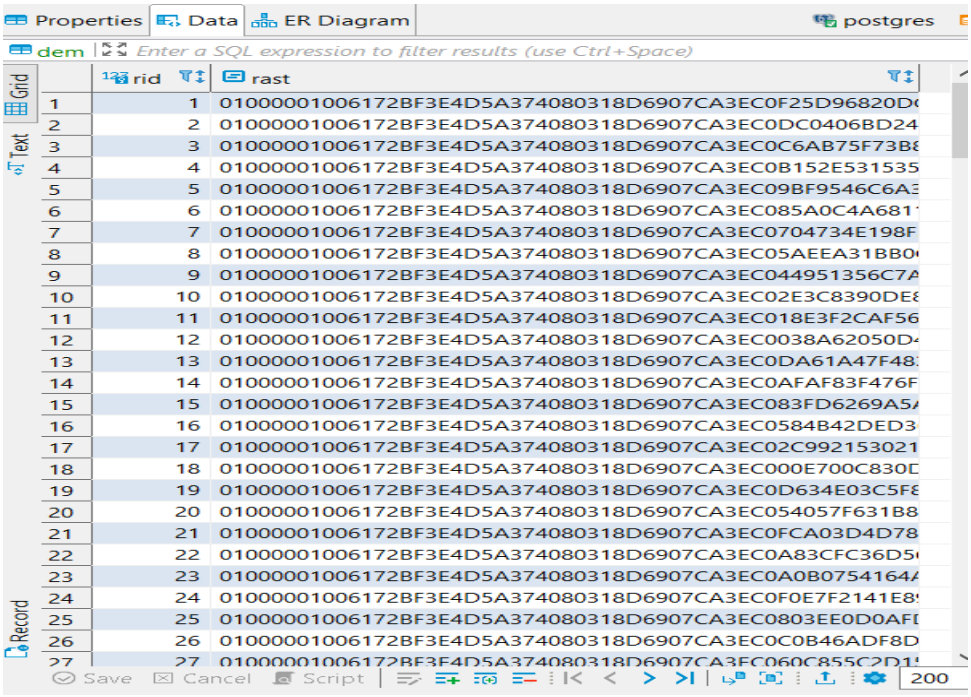
Ładowanie danych rastrowych

Przykład 1

```
C:\Program Files\PostgreSQL\14\bin>raster2pgsql -s 3763 -N -32767 -t 100x100 -I -C -M -d C:\Users\julia\Desktop\BD\cwic
enia6\srtm_1arc_v3.tif rasters.dem > C:\Users\julia\Desktop\BD\cwiczenia6\dem.sql
Processing 1/1: C:\Users\julia\Desktop\BD\cwiczenia6\srtm_1arc_v3.tif
```

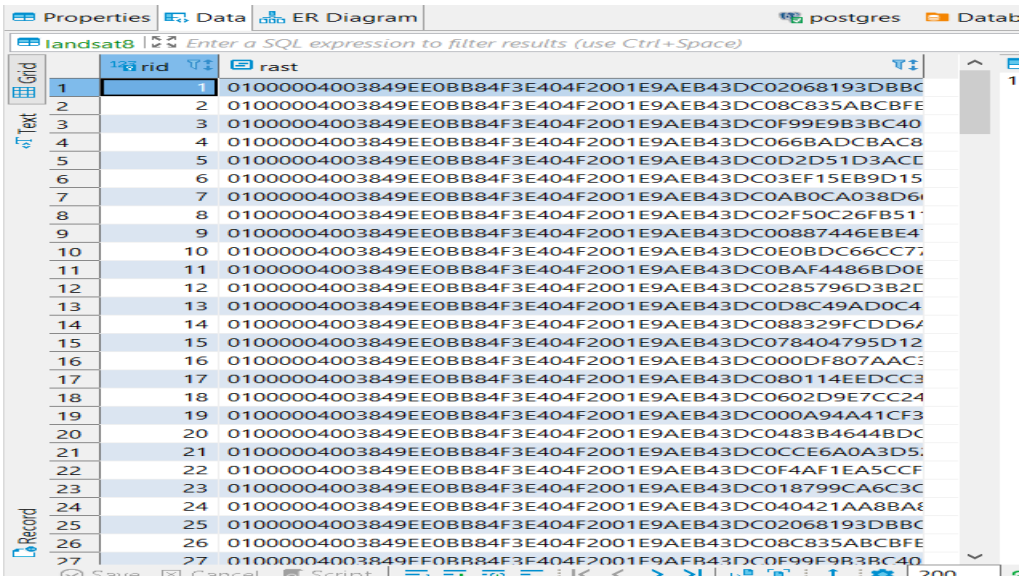
Przykład 2

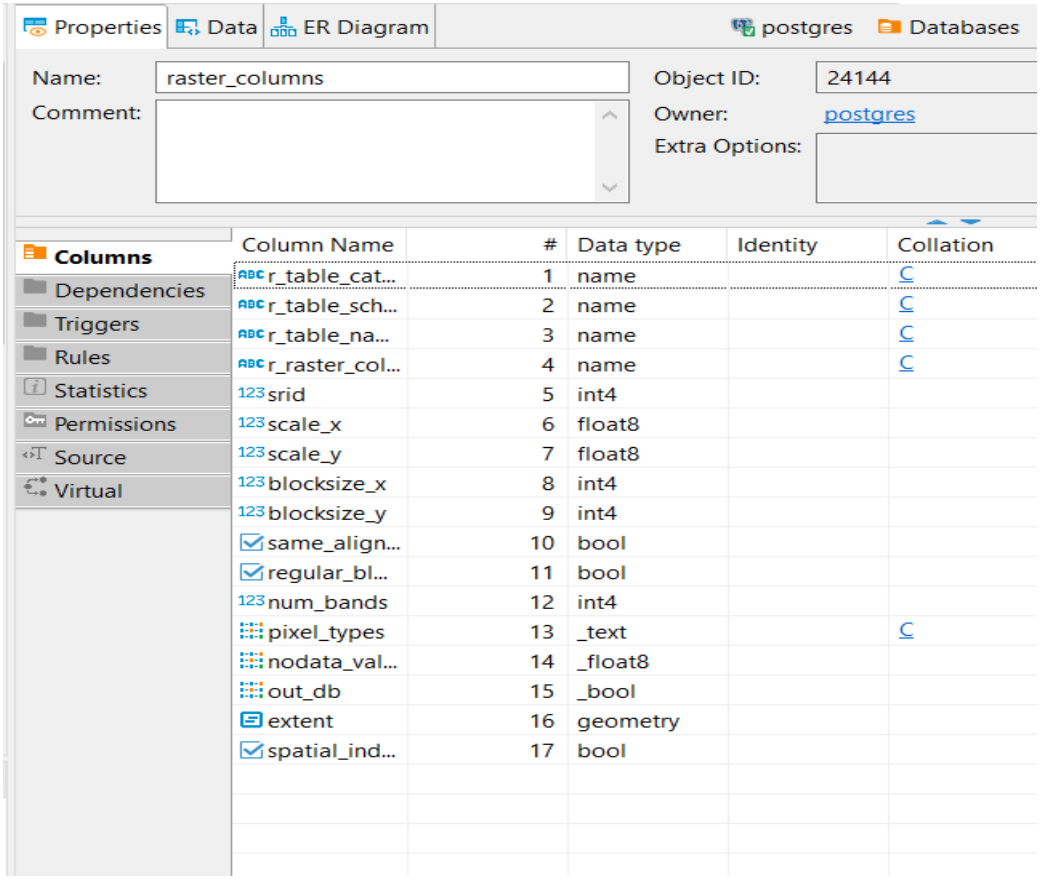
```
C:\Program Files\PostgreSQL\14\bin>raster2pgsql -s 3763 -N -32767 -t 100x100 -I -c -M -d C:\Users\julia\Desktop\BD\cwi
enia6\srtm_1arc_v3.tif rasters.dem | psql -d cwiczenia6 -h localhost -U postgres -p 5432
```

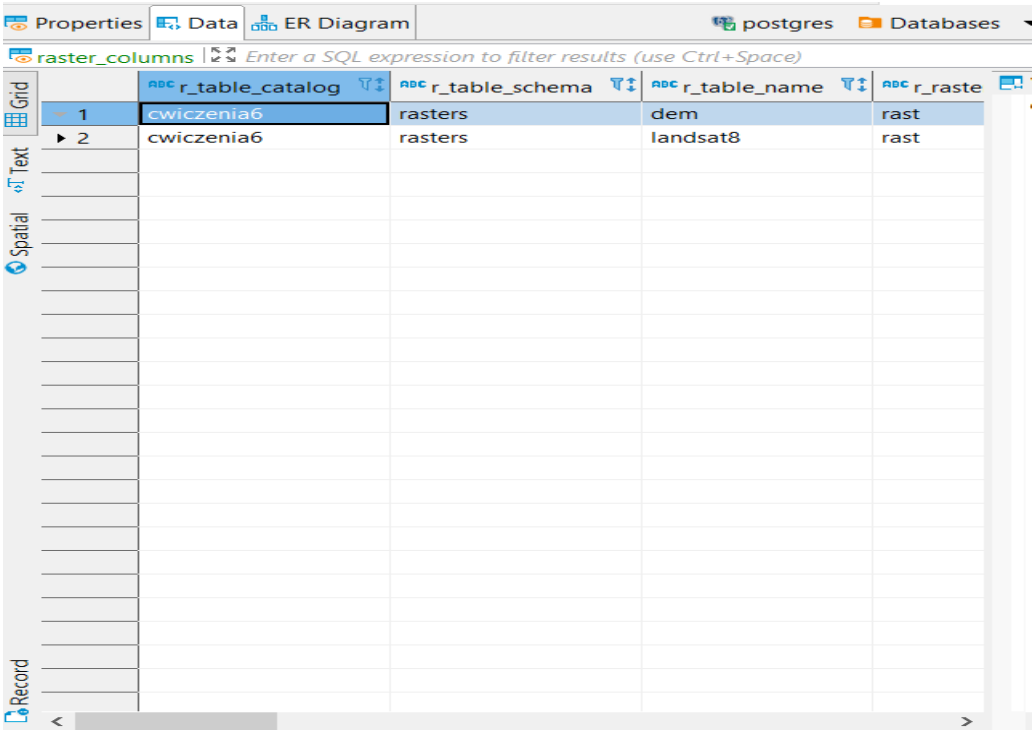


Przykład 3

```
C:\Program Files\PostgreSQL\14\bin>raster2pgsql -s 3763 -N -32767 -t 128x128 -I -C -M -d C:\Users\julia\Desktop\BD\cwicz
enia6\Landsat8_L1TP_RGBN.TIF rasters.landsat8 | psql -d cwiczenia6 -h localhost -U postgres -p 5432
```




Properties
Data
ER Diagram
postgres
Databases
Name:
raster_columns
Comment:
Object ID:
24144
Owner:
postgres
Extra Options:
Columns
Dependencies
Triggers
Rules
Statistics
Permissions
Source
Virtual
Column Name
#
Data type
Identity
Collation
r_table_cat...
1
name
C
r_table_sch...
2
name
C
r_table_na...
3
name
C
r_raster_col...
4
name
C
srid
5
int4
scale_x
6
float8
scale_y
7
float8
blocksize_x
8
int4
blocksize_y
9
int4
same_align...
10
bool
regular_bl...
11
bool
num_bands
12
int4
pixel_types
13
_text
C
nodata_val...
14
_float8
out_db
15
_bool
extent
16
geometry
spatial_ind...
17
bool


Properties
Data
ER Diagram
postgres
Databases
raster_columns
Enter a SQL expression to filter results (use Ctrl+Space)
Grid
Text
Spatial
Record
r_table_catalog
r_table_schema
r_table_name
r_raste
1
cwiczenia6
rasters
dem
rast
2
cwiczenia6
rasters
landsat8
rast

Tworzenie rastrów z istniejących rastrów i interakcja z wektorami

Przykład 1

```
--Tworzenie rastrów z istniejących rastrów i interakcja z wektorami--
--Przykład 1--
CREATE TABLE "Medrek".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';
```

Statistics 1 | Statistics 2

Name	Value
Updated Rows	25
Query	CREATE TABLE "Medrek".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'
Finish time	Wed Nov 30 11:35:29 CET 2022

```
alter table "Medrek".intersects
add column rid SERIAL PRIMARY KEY;
```

Statistics 1 | Statistics 2

Name	Value
Updated Rows	0
Query	alter table "Medrek".intersects
	add column rid SERIAL PRIMARY KEY
Finish time	Wed Nov 30 11:37:46 CET 2022

```
CREATE INDEX idx_intersects_rast_gist ON "Medrek".intersects
USING gist (ST_ConvexHull(rast));
```

Statistics 1 | Statistics 2

Connection: postgres
Time: 2022-11-30 11:38:17.588
Query: CREATE INDEX idx_intersects_rast_gist ON "Medrek".inte
USING gist (ST_ConvexHull(rast))

Name	Value
Updated Rows	0
Query	CREATE INDEX idx_intersects_rast_gist ON "Medrek".intersects
	USING gist (ST_ConvexHull(rast))
Finish time	Wed Nov 30 11:38:17 CET 2022

```
--Tworzenie rastrów z istniejących rastrów i interakcja
--Przykład 1--
CREATE TABLE "Medrek".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 i

alter table "Medrek".intersects
add column rid SERIAL PRIMARY KEY;

CREATE INDEX idx_intersects_rast_gist ON "Medrek".inters
USING gist (ST_ConvexHull(rast));

SELECT AddRasterConstraints('Medrek'::name,
'intersects'::name,'rast'::name);
```

Output

```
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
```

istics 1 | Results 2

ECT AddRasterConstraints('Medrek'::n | Enter a SQL expression to filter results (use Ctrl+Space)

addrasterconstraints
[v]

Value

true

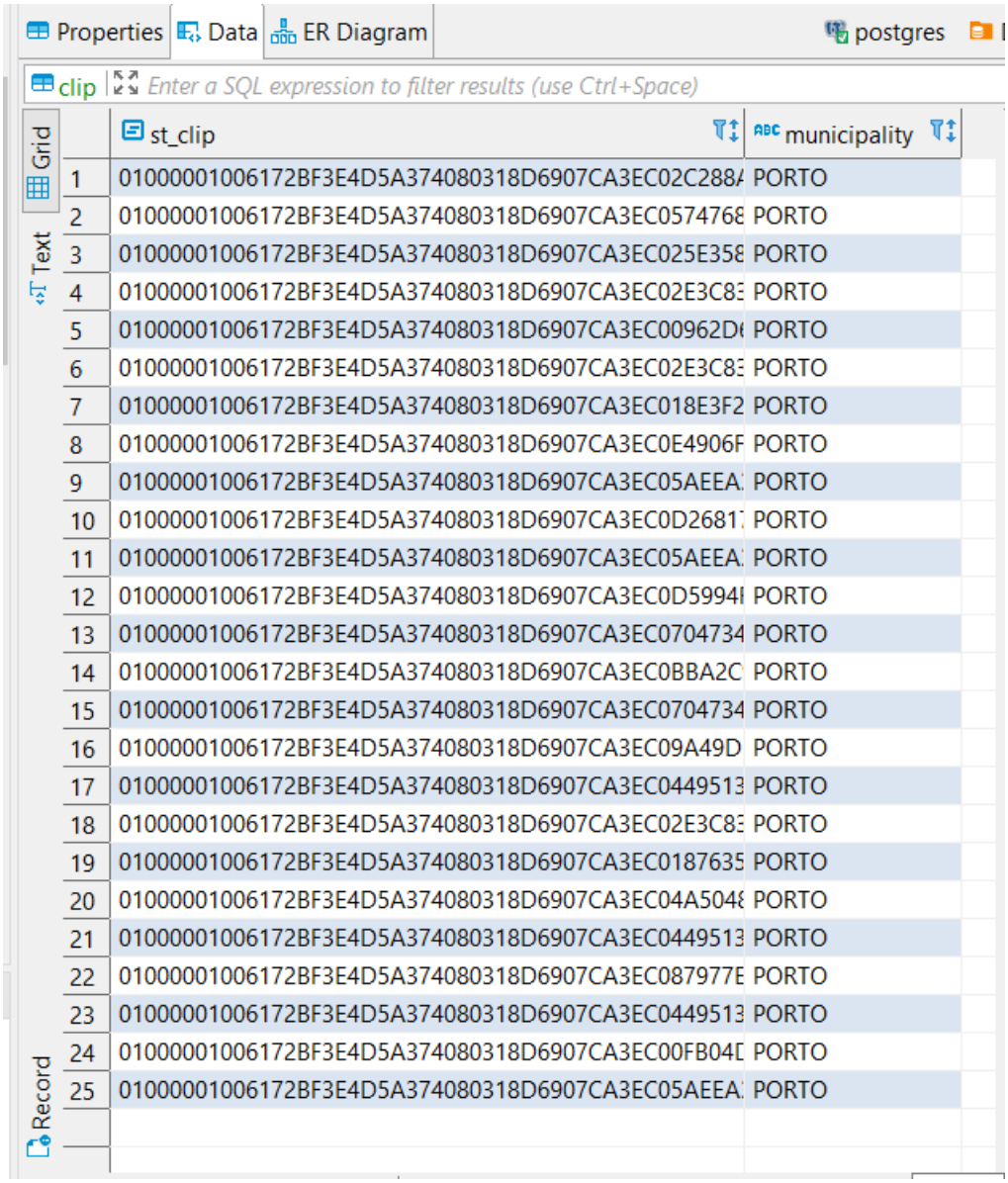
Przykład 2

```
--Przykład 2--
CREATE TABLE "Medrek".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';
```

Statistics 1 | Statistics 2

Value

Name	Value
Updated Rows	25
Query	CREATE TABLE "Medrek".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'
Finish time	Wed Nov 30 11:40:06 CET 2022



	st_clip	municipality
1	01000001006172BF3E4D5A374080318D6907CA3EC02C288A	PORTO
2	01000001006172BF3E4D5A374080318D6907CA3EC0574768	PORTO
3	01000001006172BF3E4D5A374080318D6907CA3EC025E358	PORTO
4	01000001006172BF3E4D5A374080318D6907CA3EC02E3C83	PORTO
5	01000001006172BF3E4D5A374080318D6907CA3EC00962D(	PORTO
6	01000001006172BF3E4D5A374080318D6907CA3EC02E3C83	PORTO
7	01000001006172BF3E4D5A374080318D6907CA3EC018E3F2	PORTO
8	01000001006172BF3E4D5A374080318D6907CA3EC0E4906F	PORTO
9	01000001006172BF3E4D5A374080318D6907CA3EC05AEEA1	PORTO
10	01000001006172BF3E4D5A374080318D6907CA3EC0D2681	PORTO
11	01000001006172BF3E4D5A374080318D6907CA3EC05AEEA1	PORTO
12	01000001006172BF3E4D5A374080318D6907CA3EC0D5994I	PORTO
13	01000001006172BF3E4D5A374080318D6907CA3EC0704734	PORTO
14	01000001006172BF3E4D5A374080318D6907CA3EC0BBA2C	PORTO
15	01000001006172BF3E4D5A374080318D6907CA3EC0704734	PORTO
16	01000001006172BF3E4D5A374080318D6907CA3EC09A49D	PORTO
17	01000001006172BF3E4D5A374080318D6907CA3EC0449513	PORTO
18	01000001006172BF3E4D5A374080318D6907CA3EC02E3C83	PORTO
19	01000001006172BF3E4D5A374080318D6907CA3EC0187635	PORTO
20	01000001006172BF3E4D5A374080318D6907CA3EC04A5048	PORTO
21	01000001006172BF3E4D5A374080318D6907CA3EC0449513	PORTO
22	01000001006172BF3E4D5A374080318D6907CA3EC087977E	PORTO
23	01000001006172BF3E4D5A374080318D6907CA3EC0449513	PORTO
24	01000001006172BF3E4D5A374080318D6907CA3EC00FB04D	PORTO
25	01000001006172BF3E4D5A374080318D6907CA3EC05AEEA1	PORTO

Przykład 3

```
--Przykład 3--
CREATE TABLE "Medrek".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);
```

Statistics 1 | Statistics 2

Name	Value
Updated Rows	1
Query	CREATE TABLE "Medrek".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)
Finish time	Wed Nov 30 11:41:20 CET 2022

union | Enter a SQL expression to filter results (use Ctrl+Space)

Grid | Text

	st_union
1	01000001006172BF3E4D5A374080318D6907CA3EC0D6994F2F0BE0E6C0D9E35D

Tworzenie rastrów z wektorów (rastrowanie)

Przykład 1

```
--Tworzenie rastrów z wektorów (rastrowanie)--

--Przykład 1--

CREATE TABLE "Medrek".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';
```

Statistics 1 | Statistics 2

Name	Value
Updated Rows	7
Query	CREATE TABLE "Medrek".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'
Finish time	Wed Nov 30 11:46:02 CET 2022

Properties | Data | ER Diagram | postgres

porto_parishes | Enter a SQL expression to filter results (use Ctrl+Space)

Grid | Text

	rast
1	01000001006172BF3E4D5A374080318D6907CA3EC0574768B43454E3C0D0979D
2	01000001006172BF3E4D5A374080318D6907CA3EC00A62D67B5CE2E2C08A6D9
3	01000001006172BF3E4D5A374080318D6907CA3EC0D268172037B9E5C0CA6535
4	01000001006172BF3E4D5A374080318D6907CA3EC0D6994F2F0BE0E6C05E060A
5	01000001006172BF3E4D5A374080318D6907CA3EC09A49D3957D46E4C0D9E35
6	01000001006172BF3E4D5A374080318D6907CA3EC087977E37109EE4C0A33014
7	01000001006172BF3E4D5A374080318D6907CA3EC00FB04D4A53D9E5C0474F11

Przykład 2

```
DROP TABLE "Medrek".porto_parishes;
CREATE TABLE "Medrek".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';
```

Statistics 1 | Statistics 2

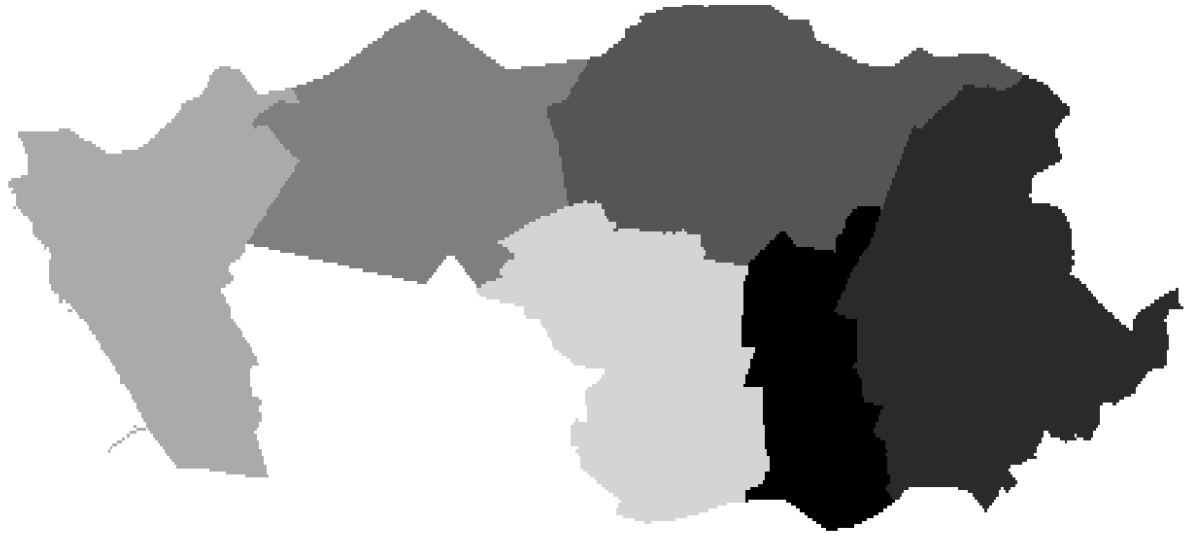
Name	Value
Updated Rows	1
Query	DROP TABLE "Medrek".porto_parishes;
	CREATE TABLE "Medrek".porto_parishes AS
	WITH r AS (
	SELECT rast FROM rasters.dem
	LIMIT 1
	)
	SELECT st_union(ST_AsRaster(a.geom,r.rast,'8BUI',a.id,-32767)) AS rast
	FROM vectors.porto_parishes AS a, r
	WHERE a.municipality ilike 'porto'
Finish time	Wed Nov 30 11:47:43 CET 2022

Properties | Data | ER Diagram — postgres

porto_parishes — Enter a SQL expression to filter results (use Ctrl+Space)

Grid / Text

	rast
1	01000001006172BF3E4D5A374080318D6907CA3EC0D6994F2F0BE0E6C0DAE35D

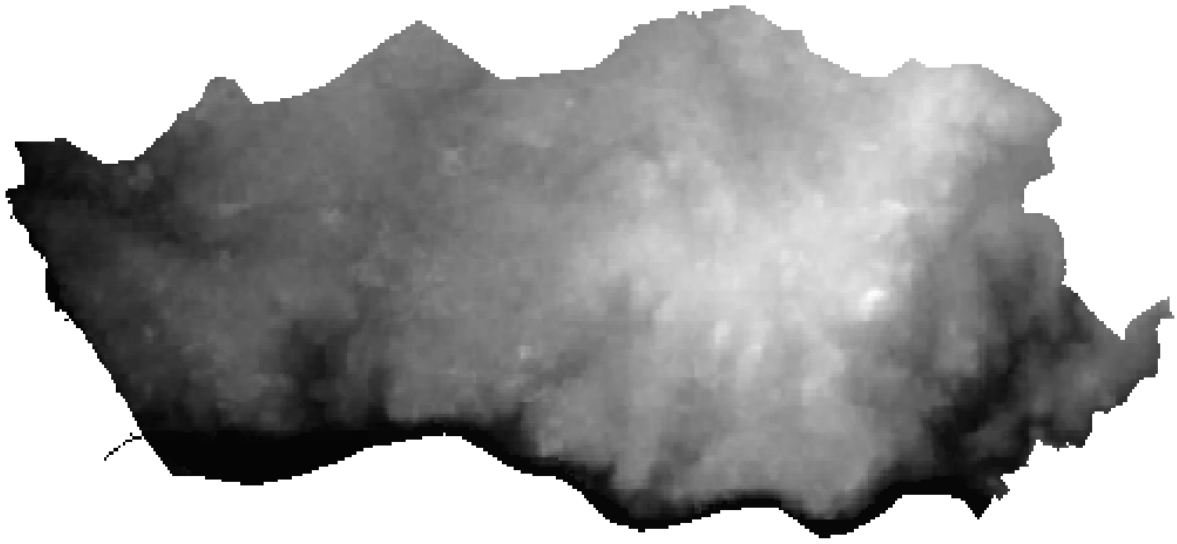


Przykład 3

```
--Przykład 3--
DROP TABLE "Medrek".porto_parishes;
CREATE TABLE "Medrek".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';
```

Statistics 1 | Statistics 2

Name	Value
Updated Rows	8
Query	DROP TABLE "Medrek".porto_parishes;
	CREATE TABLE "Medrek".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'
Finish time	Wed Nov 30 11:49:20 CET 2022



Properties | Data | ER Diagram | postgres

porto_parishes | Enter a SQL expression to filter results (use Ctrl+Space)

Grid | Text

	rast
1	0100000100617 2BF3E4D5A374080318D6907CA3EC0D6994F2F0BE0E6C0DAE35DC008960441000000
2	0100000100617 2BF3E4D5A374080318D6907CA3EC0B0A2635B666AE5C0DAE35DC008960441000000
3	0100000100617 2BF3E4D5A374080318D6907CA3EC08AAB7787C1F4E3C0DAE35DC008960441000000
4	0100000100617 2BF3E4D5A374080318D6907CA3EC064B48BB31C7FE2C0DAE35DC008960441000000
5	0100000100617 2BF3E4D5A374080318D6907CA3EC0D6994F2F0BE0E6C014AFB7A2E01A044100000
6	0100000100617 2BF3E4D5A374080318D6907CA3EC0B0A2635B666AE5C014AFB7A2E01A044100000
7	0100000100617 2BF3E4D5A374080318D6907CA3EC08AAB7787C1F4E3C014AFB7A2E01A044100000
8	0100000100617 2BF3E4D5A374080318D6907CA3EC064B48BB31C7FE2C014AFB7A2E01A044100000

Konwertowanie rastrów na wektory (wektoryzowanie)

Przykład 1

```
--Przykład 1--
create table "Medrek".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);
```

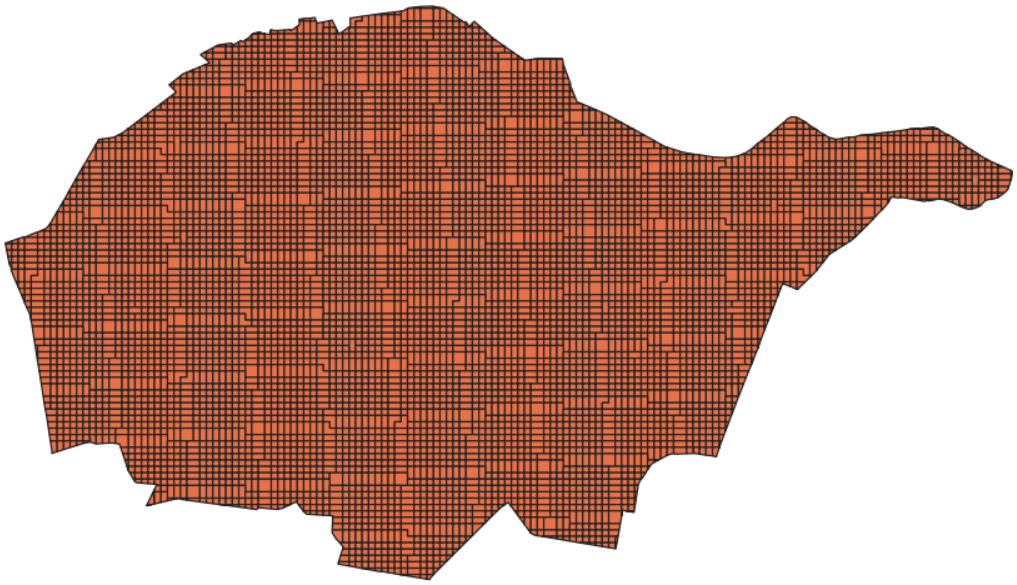
Statistics 1 | Statistics 2

Name	Value
Updated Rows	6629
Query	--Przykład 1--
	create table "Medrek".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)
Finish time	Wed Nov 30 11:53:22 CET 2022

Properties | Data | ER Diagram — postgres

intersection — Enter a SQL expression to filter results (use Ctrl+Space)

	rid	geom	val
1	221	POLYGON ((-39604.86528035818 168624.02902255123, -39633.58795484908 1	10,648
2	221	POLYGON ((-39574.553878279854 168624.02902255123, -39604.86528035818	12,155
3	221	POLYGON ((-39786.73369282809 168594.3232286338, -39794.43786661896 16	9,248
4	221	POLYGON ((-39756.42229074977 168594.3232286338, -39786.73369282809 16	10,030
5	221	POLYGON ((-39726.11088867145 168594.3232286338, -39756.42229074977 16	10,347
6	221	POLYGON ((-39695.79948659313 168594.3232286338, -39726.11088867145 16	10,126
7	221	POLYGON ((-39665.488084514815 168594.3232286338, -39695.79948659313 1	10,611
8	221	POLYGON ((-39635.1766824365 168564.61743471635, -39665.488084514815 1	12,761
9	221	POLYGON ((-39604.86528035818 168624.02902255123, -39604.86528035818 1	14,382
10	221	POLYGON ((-39574.553878279854 168624.02902255123, -39574.55387827985	14,090
11	221	POLYGON ((-40120.159115689596 168564.61743471635, -40140.8828966886 1	9,124
12	221	POLYGON ((-40089.84771361128 168564.61743471635, -40120.159115689596	10,759
13	221	POLYGON ((-40089.84771361128 168564.61743471635, -40089.84771361128 1	10,625
14	221	POLYGON ((-39998.913507376325 168534.9116407989, -40029.22490945464 1	9,606
15	221	POLYGON ((-39998.913507376325 168534.9116407989, -39998.913507376325	8,699
16	221	POLYGON ((-39907.979301141364 168534.9116407989, -39912.205874024134	8,290
17	221	POLYGON ((-39847.35649698473 168564.61743471635, -39877.667899063046	8,601
18	221	POLYGON ((-39786.73369282809 168594.3232286338, -39786.73369282809 16	10,102
19	221	POLYGON ((-39756.42229074977 168594.3232286338, -39756.42229074977 16	11,050
20	221	POLYGON ((-39726.11088867145 168594.3232286338, -39726.11088867145 16	10,046
21	221	POLYGON ((-39695.79948659313 168594.3232286338, -39695.79948659313 16	9,908
22	221	POLYGON ((-39665.488084514815 168594.3232286338, -39665.488084514815	9,186
23	221	POLYGON ((-39817.04509490641 168534.9116407989, -39877.667899063046 1	9,102
24	221	POLYGON ((-39635.1766824365 168564.61743471635, -39635.1766824365 168	9,849
25	221	POLYGON ((-39604.86528035818 168564.61743471635, -39604.86528035818 1	13,518
26	221	POLYGON ((-39574.553878279854 168564.61743471635, -39574.55387827985	13,258
27	221	POLYGON ((-40332.33893023783 168505.20584688146, -40358.59956761706 1	11,353

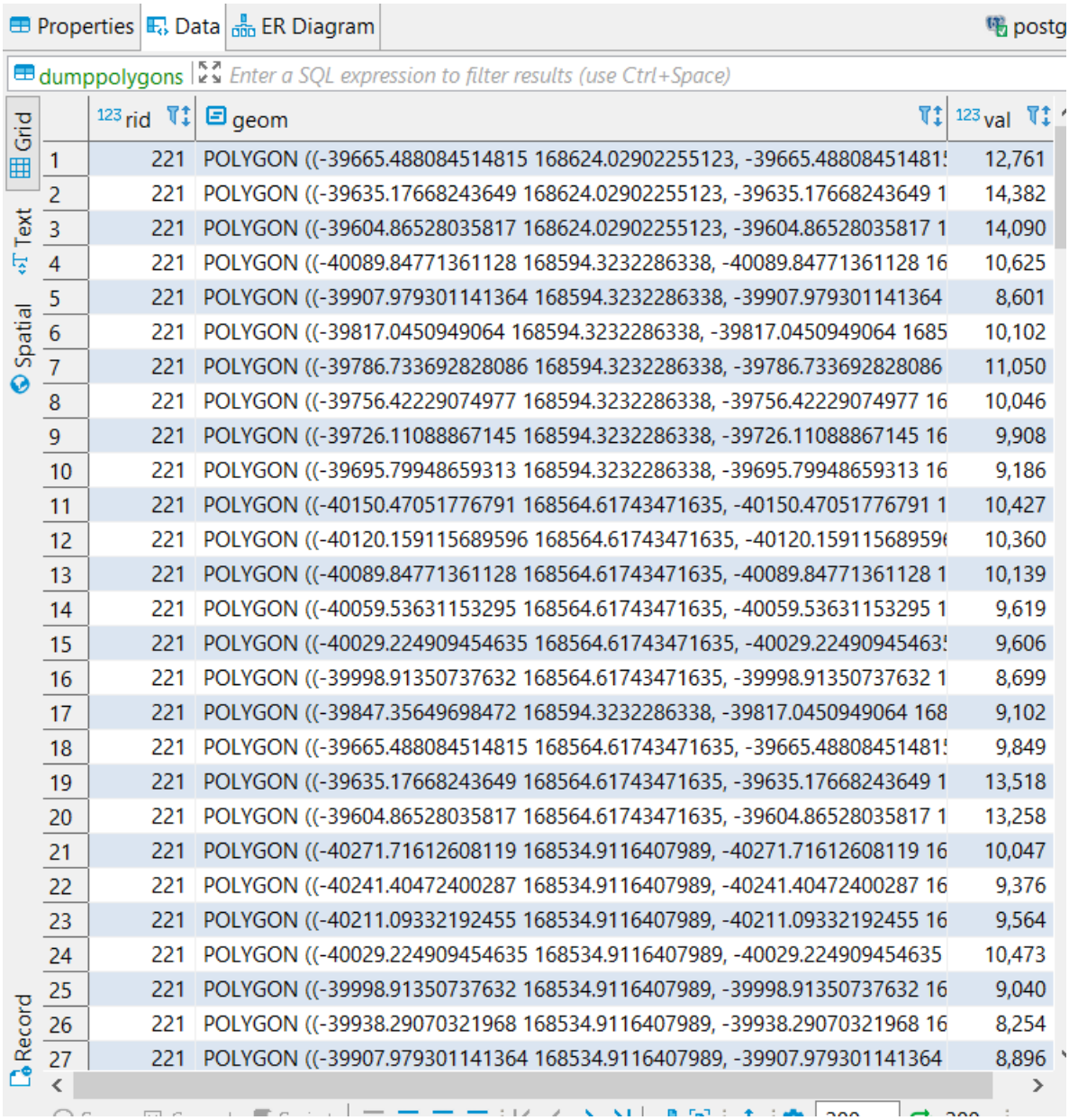


Przykład 2

```
--Przykład 2--
CREATE TABLE "Medrek".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);
```

Statistics 1 | Statistics 2

me	Value
dated Rows	6422
ery	--Przykład 2--
	CREATE TABLE "Medrek".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)
ish time	Wed Nov 30 11:54:37 CET 2022



	rid	geom	val
1	221	POLYGON ((-39665.488084514815 168624.02902255123, -39665.48808451481	12,761
2	221	POLYGON ((-39635.17668243649 168624.02902255123, -39635.17668243649 1	14,382
3	221	POLYGON ((-39604.86528035817 168624.02902255123, -39604.86528035817 1	14,090
4	221	POLYGON ((-40089.84771361128 168594.3232286338, -40089.84771361128 16	10,625
5	221	POLYGON ((-39907.979301141364 168594.3232286338, -39907.979301141364	8,601
6	221	POLYGON ((-39817.0450949064 168594.3232286338, -39817.0450949064 1685	10,102
7	221	POLYGON ((-39786.733692828086 168594.3232286338, -39786.733692828086	11,050
8	221	POLYGON ((-39756.42229074977 168594.3232286338, -39756.42229074977 16	10,046
9	221	POLYGON ((-39726.11088867145 168594.3232286338, -39726.11088867145 16	9,908
10	221	POLYGON ((-39695.79948659313 168594.3232286338, -39695.79948659313 16	9,186
11	221	POLYGON ((-40150.47051776791 168564.61743471635, -40150.47051776791 1	10,427
12	221	POLYGON ((-40120.159115689596 168564.61743471635, -40120.15911568959	10,360
13	221	POLYGON ((-40089.84771361128 168564.61743471635, -40089.84771361128 1	10,139
14	221	POLYGON ((-40059.53631153295 168564.61743471635, -40059.53631153295 1	9,619
15	221	POLYGON ((-40029.224909454635 168564.61743471635, -40029.22490945463	9,606
16	221	POLYGON ((-39998.91350737632 168564.61743471635, -39998.91350737632 1	8,699
17	221	POLYGON ((-39847.35649698472 168594.3232286338, -39817.0450949064 168	9,102
18	221	POLYGON ((-39665.488084514815 168564.61743471635, -39665.48808451481	9,849
19	221	POLYGON ((-39635.17668243649 168564.61743471635, -39635.17668243649 1	13,518
20	221	POLYGON ((-39604.86528035817 168564.61743471635, -39604.86528035817 1	13,258
21	221	POLYGON ((-40271.71612608119 168534.9116407989, -40271.71612608119 16	10,047
22	221	POLYGON ((-40241.40472400287 168534.9116407989, -40241.40472400287 16	9,376
23	221	POLYGON ((-40211.09332192455 168534.9116407989, -40211.09332192455 16	9,564
24	221	POLYGON ((-40029.224909454635 168534.9116407989, -40029.224909454635	10,473
25	221	POLYGON ((-39998.91350737632 168534.9116407989, -39998.91350737632 16	9,040
26	221	POLYGON ((-39938.29070321968 168534.9116407989, -39938.29070321968 16	8,254
27	221	POLYGON ((-39907.979301141364 168534.9116407989, -39907.979301141364	8,896

Analiza rastrów

Przykład 1

```sql
--Przykład 1

CREATE TABLE "Medrek".landsat_nir AS
SELECT rid, ST_Band(rast,4) AS rast
FROM rasters.landsat8;
```

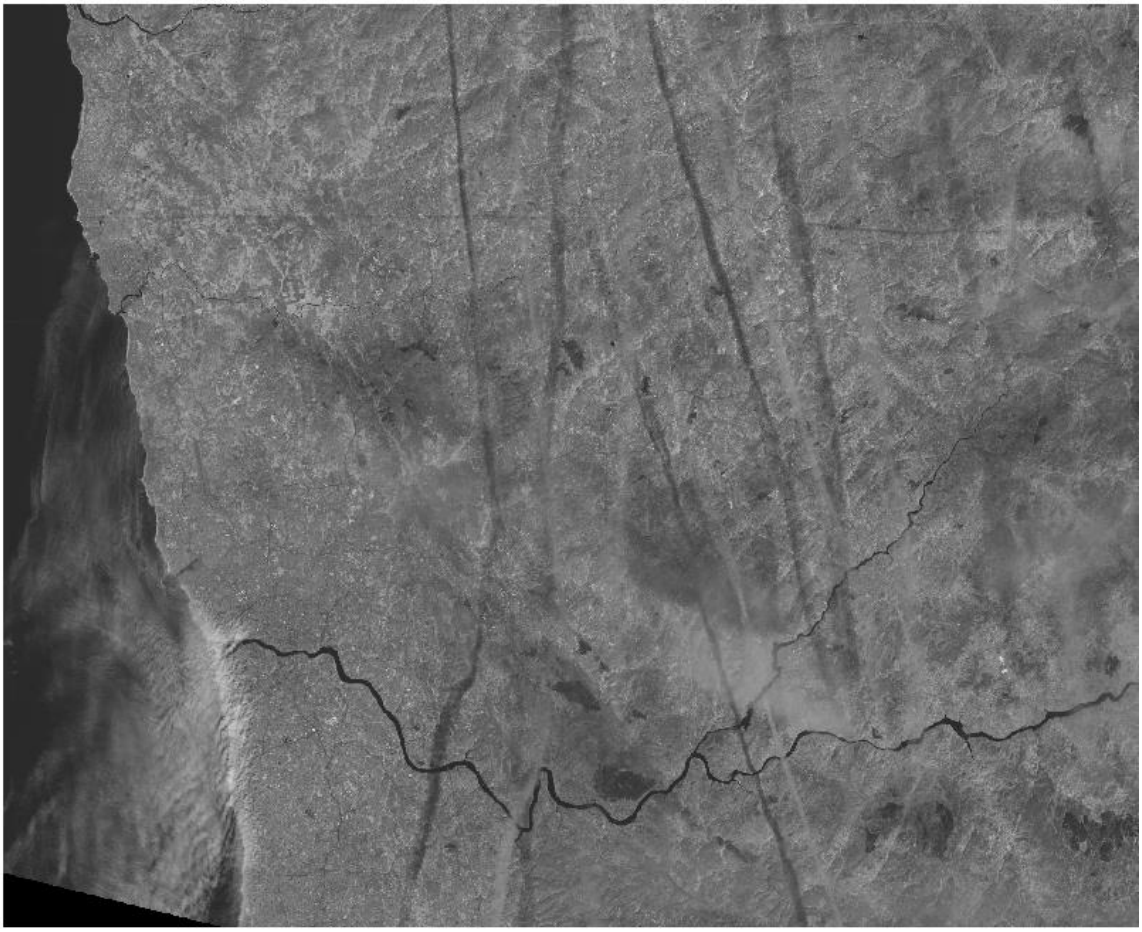
Statistics 1 | Statistics 2

Name	Value
Updated Rows	384
	CREATE TABLE "Medrek".landsat_nir AS
	SELECT rid, ST_Band(rast,4) AS rast
	FROM rasters.landsat8
Finish time	Wed Nov 30 11:57:35 CET 2022

landsat_nir | Enter a SQL expression to filter results (use Ctrl+Space)

Grid | Text | Record

	rid	rast
1	1	0100000100 3849EE0BB84F3E404F2001E9AEB43DC02068193DBBCBECC071695283BC2C0
2	2	0100000100 3849EE0BB84F3E404F2001E9AEB43DC08C835ABCBFE6EAC071695283BC2C0
3	3	0100000100 3849EE0BB84F3E404F2001E9AEB43DC0F99E9B3BC401E9C071695283BC2C09
4	4	0100000100 3849EE0BB84F3E404F2001E9AEB43DC066BADCBAC81CE7C071695283BC2C0
5	5	0100000100 3849EE0BB84F3E404F2001E9AEB43DC0D2D51D3ACD37E5C071695283BC2C0
6	6	0100000100 3849EE0BB84F3E404F2001E9AEB43DC03EF15EB9D152E3C071695283BC2C09
7	7	0100000100 3849EE0BB84F3E404F2001E9AEB43DC0AB0CA038D66DE1C071695283BC2C0
8	8	0100000100 3849EE0BB84F3E404F2001E9AEB43DC02F50C26FB511DFC071695283BC2C09
9	9	0100000100 3849EE0BB84F3E404F2001E9AEB43DC00887446EBE47DBC071695283BC2C0
10	10	0100000100 3849EE0BB84F3E404F2001E9AEB43DC0E0BDC66CC77DD7C071695283BC2C0
11	11	0100000100 3849EE0BB84F3E404F2001E9AEB43DC0BAF4486BD0B3D3C071695283BC2C0
12	12	0100000100 3849EE0BB84F3E404F2001E9AEB43DC0285796D3B2D3CFC071695283BC2C0
13	13	0100000100 3849EE0BB84F3E404F2001E9AEB43DC0D8C49AD0C43FC8C071695283BC2C0
14	14	0100000100 3849EE0BB84F3E404F2001E9AEB43DC088329FCDD6ABC0C071695283BC2C0
15	15	0100000100 3849EE0BB84F3E404F2001E9AEB43DC078404795D12FB2C071695283BC2C09
16	16	0100000100 3849EE0BB84F3E404F2001E9AEB43DC000DF807AAC3F88C071695283BC2C0
17	17	0100000100 3849EE0BB84F3E404F2001E9AEB43DC080114EEDCC3FA84071695283BC2C0
18	18	0100000100 3849EE0BB84F3E404F2001E9AEB43DC0602D9E7CC247BB4071695283BC2C0
19	19	0100000100 3849EE0BB84F3E404F2001E9AEB43DC000A94A41CF37C54071695283BC2C0
20	20	0100000100 3849EE0BB84F3E404F2001E9AEB43DC0483B4644BDCBCC4071695283BC2C0
21	21	0100000100 3849EE0BB84F3E404F2001E9AEB43DC0CCE6A0A3D52FD24071695283BC2C0
22	22	0100000100 3849EE0BB84F3E404F2001E9AEB43DC0F4AF1EA5CCF9D54071695283BC2C0
23	23	0100000100 3849EE0BB84F3E404F2001E9AEB43DC018799CA6C3C3D94071695283BC2C0
24	24	0100000100 3849EE0BB84F3E404F2001E9AEB43DC040421AA8BA8DDD4071695283BC2C0
25	25	0100000100 3849EE0BB84F3E404F2001E9AEB43DC02068193DBBCBECC0F064AEC7E9B50
26	26	0100000100 3849EE0BB84F3E404F2001E9AEB43DC08C835ABCBFE6EAC0F064AEC7E9B508
27	27	0100000100 3849EE0BB84F3E404F2001E9AEB43DC0F99E9B3BC401E9C0F064AEC7E9B508



Przykład 2

```
--Przykład 2

CREATE TABLE "Medrek".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);
```

Statistics 1 | Statistics 2

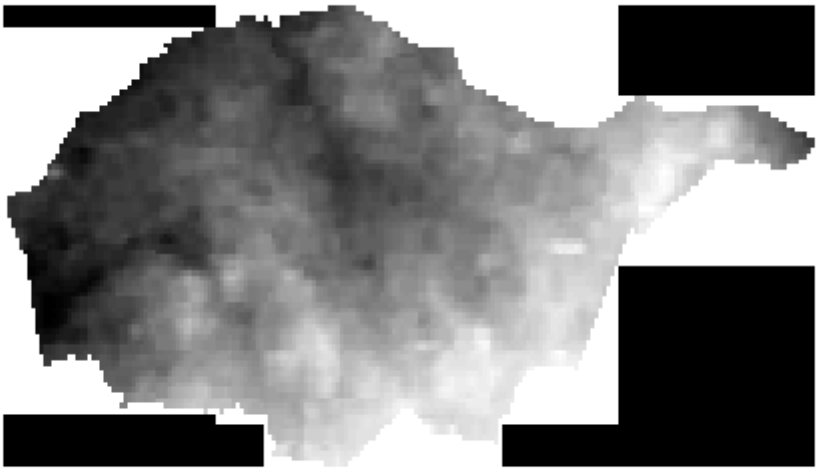
Name	Value
Updated Rows	4
Query	CREATE TABLE "Medrek".paranhos_dem AS
	SELECT a.rid,ST_Clip(a.rast, b.geom,true) as rast
	FROM rasters.dem AS a, vectors.porto_parishes AS b
	WHERE b.parish ilike 'paranhos' and ST_Intersects(b.geom,a.rast)
Finish time	Wed Nov 30 12:02:35 CET 2022

Properties | Data | ER Diagram | postgres

paranhos_dem | Enter a SQL expression to filter results (use Ctrl+Space)

Grid | Text

	rid	rast
1	380	01000001006172BF3E4D5A374080318D6907CA3EC09A49D3957D46E4C033B2707F2F920
2	381	01000001006172BF3E4D5A374080318D6907CA3EC044951356C7ABE3C0DAE35DC00896(
3	382	01000001006172BF3E4D5A374080318D6907CA3EC02E3C8390DE87E2C0D7D06D6CAD85
4	412	01000001006172BF3E4D5A374080318D6907CA3EC0187635E2BF88E3C0474F11FE054A04



```
--Przykład 3

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

Statistics 1 | Statistics 2

Name	Value
Updated Rows	4
Query	CREATE TABLE "Medrek".paranhos_slope AS
	SELECT a.rid,ST_Slope(a.rast,1,'32BF','PERCENTAGE') as rast
	FROM "Medrek".paranhos_dem AS a
Finish time	Wed Nov 30 12:03:14 CET 2022

Properties | Data | ER Diagram | postgre

paranhos_slope | Enter a SQL expression to filter results (use Ctrl+Space)

Grid | Text

	rid	rast
1	380	01000001006172BF3E4D5A374080318D6907CA3EC09A49D3957D46E4C033B2707F2F920
2	381	01000001006172BF3E4D5A374080318D6907CA3EC044951356C7ABE3C0DAE35DC00896(
3	382	01000001006172BF3E4D5A374080318D6907CA3EC02E3C8390DE87E2C0D7D06D6CAD85
4	412	01000001006172BF3E4D5A374080318D6907CA3EC0187635E2BF88E3C0474F11FE054A04

Przykład 4

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

Statistics 1 | Statistics 2 | Statistics 3

Name	Value
Updated Rows	4
Query	CREATE TABLE "Medrek".paranhos_slope_reclass AS
	SELECT a.rid,ST_Reclass(a.rast,1,']0-15]:1, (15-30]:2, (30-9999:3',
	'32BF',0)
	FROM "Medrek".paranhos_slope AS a
Finish time	Wed Nov 30 12:05:01 CET 2022

Properties | Data | ER Diagram — postgres — Databas

paranhos_slope_reclass — Enter a SQL expression to filter results (use Ctrl+Space)

Grid / Text

	rid	st_reclass
1	380	01000001006172BF3E4D5A374080318D6907CA3EC09A49D3957D46E4C033B2707F2F920
2	381	01000001006172BF3E4D5A374080318D6907CA3EC044951356C7ABE3C0DAE35DC00896(
3	382	01000001006172BF3E4D5A374080318D6907CA3EC02E3C8390DE87E2C0D7D06D6CAD85
4	412	01000001006172BF3E4D5A374080318D6907CA3EC0187635E2BF88E3C0474F11FE054A04

Przykład 5

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

Statistics 1 | Statistics 2 | Results 3

SELECT st_summarystats(a.rast) AS stats FROM "M — Enter a SQL expression to filter results (use Ctrl+Space)

	stats					
	count	sum	mean	stddev	min	max
1	2,616	278,385	106.4162844037	11.6226287622	87	143
2	6,463	816,615	126.3523131673	14.0438229209	94	158
3	682	95,581	140.1480938416	12.0780721866	103	158
4	216	31,874	147.5648148148	4.2628306283	137	158

Przykład 6

```
--Przykład 6
SELECT st_summarystats(ST_Union(a.rast))
FROM "Medrek".paranhos_dem AS a;
```

Statistics 1 | Statistics 2 | Results 3

SELECT st_summarystats(ST_Union(a.rast)) FROM — Enter a SQL expression to filter results (use Ctrl+Spac

	st_summarystats					
	count	sum	mean	stddev	min	max
1	9,977	1,222,455	122.5273128195	16.9080042027	87	158

Przykład 7

```
--Przykład 7
WITH t AS (
SELECT st_summarystats(ST_Union(a.rast)) AS stats
FROM "Medrek".paranhos_dem AS a
)
SELECT (stats).min,(stats).max,(stats).mean FROM t;
```

tatistics 1 | Statistics 2 | Results 3

VITH t AS (SELECT st_summarystats(ST_Union(a.r — Enter a SQL expression to filter results (use Ct

	min	max	mean
1	87	158	122.5273128195

Przykład 8

```
--Przykład 8
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;
```

tatistics 1 | Statistics 2 | porto_parishes 3 ×

VITH t AS (SELECT b.parish AS parish, st_summar | Enter a SQL expression to filter results (use Ctrl+Space)

	parish	min	max	mean
1	Bonfim	1	159	107.5658842668
2	Campanhã	0	178	74.6673221309
3	Paranhos	87	158	122.5273128195
4	Ramalde	48	108	77.5844444444
5	União das freguesias de Aldoar, Foz do Douro e Nevogild	-4	83	34.6673548979
6	União das freguesias de Cedofeita, Santo Ildefonso, Sé, M	1	157	95.0027774104
7	União das freguesias de Lordelo do Ouro e Massarelos	-1	117	49.5005144033

Przykład 9

```
--Przykład 9
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;
```

Statistics 1 | Statistics 2 | places 3 ×

SELECT b.name,st_value(a.rast,(ST_Dump(b.geom | Enter a SQL expression to filter results (use Ctrl+Space)

Grid | Text | cord

	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
12	Felgueiras	320
13	Gondomar	123
14	Guifões	69
15	Guimarães	197
16	Lousada	289

Przykład 10

```
--Przykład 10
create table "Medrek".tpi30 as
select ST_TPI(a.rast,1) as rast
from rasters.dem a;

CREATE INDEX idx_tpi30_rast_gist ON "Medrek".tpi30
USING gist (ST_ConvexHull(rast));

SELECT AddRasterConstraints('Medrek'::name,
'tpi30'::name,'rast'::name);
```

tatistics 1 | Statistics 2 | Results 3 ×

reate table "Medrek".tpi30 as select ST_TPI(a.rast | Enter a SQL expression to filter re

	addrasterconstraints
1	[v]

1 row(s) fetched - 1m 26s, on 202

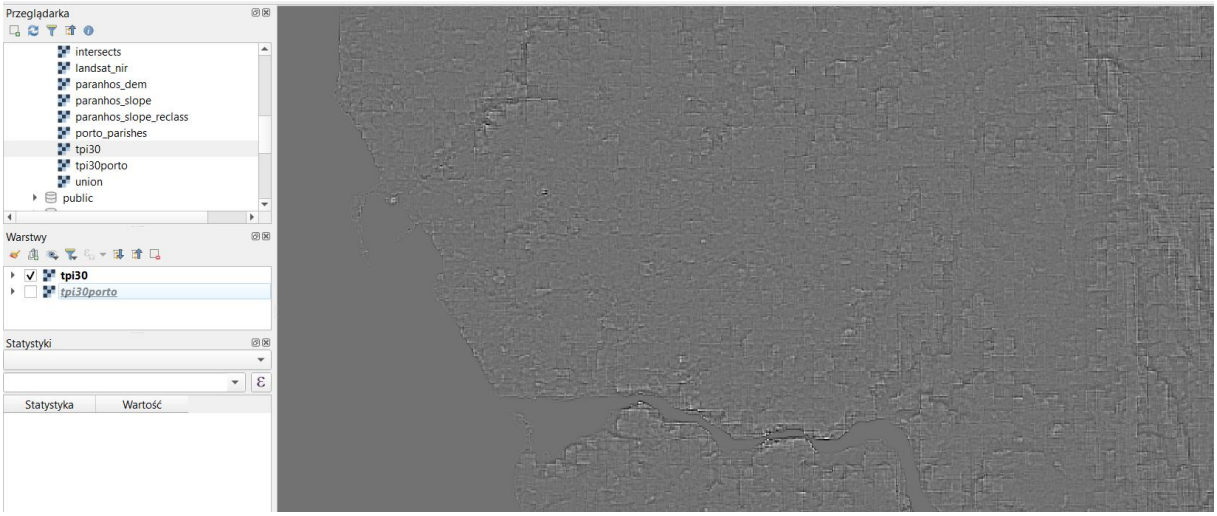
```
--Przykład 10 tylko dla porto
create table "Medrek".tpi30porto as
select ST_TPI(a.rast,1) as rast
from rasters.dem a,vectors.porto_parishes AS b
where ST_Intersects(a.rast, b.geom) AND b.municipality ilike 'porto';

CREATE INDEX idx_tpi30_rast_gist_porto ON "Medrek".tpi30porto
USING gist (ST_ConvexHull(rast));

SELECT AddRasterConstraints('Medrek'::name,
'tpi30porto'::name,'rast'::name);
```

stics 1 | Statistics 2 | Results 3 ×

te table "Medrek".tpi30porto as select ST_TPI | Enter a SQL expression to filter results (use Ctrl+Spa

addrasterconstraints
[v]

1 row(s) fetched - 4.71s,



Algebra map

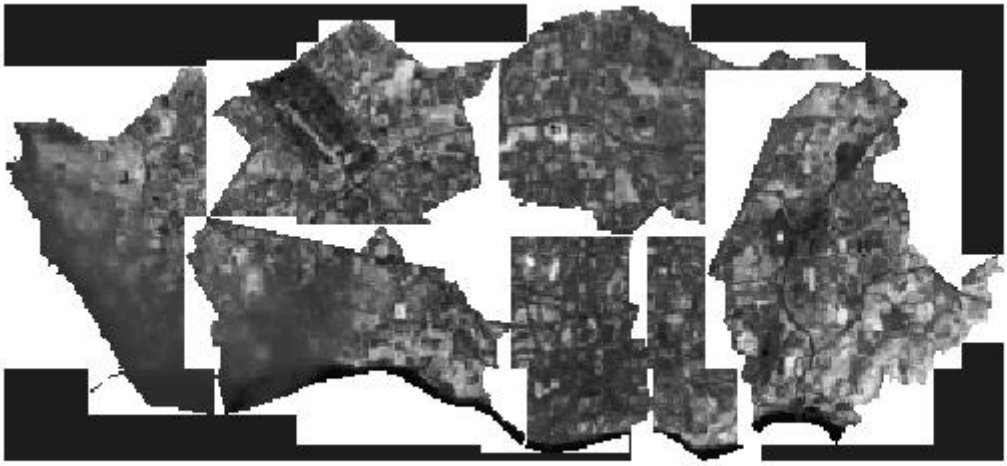
Przykład 1

```sql
--Przykład 1
CREATE TABLE "Medrek".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 "Medrek".porto_ndvi
USING gist (ST_ConvexHull(rast));

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

porto_ndvi | Enter a SQL expression to filter results (use Ctrl+Space)

Grid | Text

	rid	rast
1	245	0100000100 3849EE0BB84F3E404F2001E9AEB43DC0086F60B09B56E3C01F1195122C32044
2	246	0100000100 3849EE0BB84F3E404F2001E9AEB43DC03EF15EB9D152E3C075C249A2725504
3	270	0100000100 3849EE0BB84F3E404F2001E9AEB43DC075735DC2074FE3C0E4374673AE1104
4	246	0100000100 3849EE0BB84F3E404F2001E9AEB43DC075B133BEF0E4E2C00224CA124B7E04
5	247	0100000100 3849EE0BB84F3E404F2001E9AEB43DC0AB0CA038D66DE1C079E13D89743BC
6	270	0100000100 3849EE0BB84F3E404F2001E9AEB43DC03E4E2033AFB3E2C0E4374673AE1104
7	221	0100000100 3849EE0BB84F3E404F2001E9AEB43DC0C069108BB315E5C0B6C41B00DC9004
8	244	0100000100 3849EE0BB84F3E404F2001E9AEB43DC08A8A5008A0B8E5C02644A7A801820
9	245	0100000100 3849EE0BB84F3E404F2001E9AEB43DC0D2D51D3ACD37E5C0653CEA2E8188C
10	244	0100000100 3849EE0BB84F3E404F2001E9AEB43DC0D4DDC44A29E0E6C014B4B85D26800
11	268	0100000100 3849EE0BB84F3E404F2001E9AEB43DC0AF24956AEA66E6C0E4374673AE1104
12	221	0100000100 3849EE0BB84F3E404F2001E9AEB43DC0E31A995A96E6E3C0EC74E7E06D9604
13	222	0100000100 3849EE0BB84F3E404F2001E9AEB43DC03EF15EB9D152E3C0EC74E7E06D9604
14	245	0100000100 3849EE0BB84F3E404F2001E9AEB43DC08863BE794F45E4C0653CEA2E818804
15	246	0100000100 3849EE0BB84F3E404F2001E9AEB43DC03EF15EB9D152E3C0653CEA2E818804
16	245	0100000100 3849EE0BB84F3E404F2001E9AEB43DC09BB0E0AA749CE4C07E0AC14760560
17	246	0100000100 3849EE0BB84F3E404F2001E9AEB43DC03EF15EB9D152E3C0C121F8B4E14204
18	269	0100000100 3849EE0BB84F3E404F2001E9AEB43DC008F3B3B8C92AE4C0E4374673AE1104
19	270	0100000100 3849EE0BB84F3E404F2001E9AEB43DC03EF15EB9D152E3C0E4374673AE1104
20	244	0100000100 3849EE0BB84F3E404F2001E9AEB43DC09CF65DB7B9DAE5C0001A3B3B61490
21	245	0100000100 3849EE0BB84F3E404F2001E9AEB43DC0D2D51D3ACD37E5C0E541D54A9846
22	268	0100000100 3849EE0BB84F3E404F2001E9AEB43DC02E114C2342ADE5C0E4374673AE1104
23	269	0100000100 3849EE0BB84F3E404F2001E9AEB43DC0D2D51D3ACD37E5C0E4374673AE110



Przykład 2

```
create or replace function "Medrek".ndvi(
value double precision [] [] [],
pos integer [][],
VARIADIC userargs text []
)
RETURNS double precision AS
$$
BEGIN

RETURN (value [2][1][1] - value [1][1][1])/(value [2][1][1]+value
[1][1][1]);
END;
$$
LANGUAGE 'plpgsql' IMMUTABLE COST 1000;


CREATE TABLE "Medrek".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],
'"Medrek".ndvi(double precision[],
integer[],text[])'::regprocedure, --> This is the function!
'32BF'::text
) AS rast
FROM r;

CREATE INDEX idx_porto_ndvi2_rast_gist ON "Medrek".porto_ndvi2
```

Properties | Data | ER Diagram

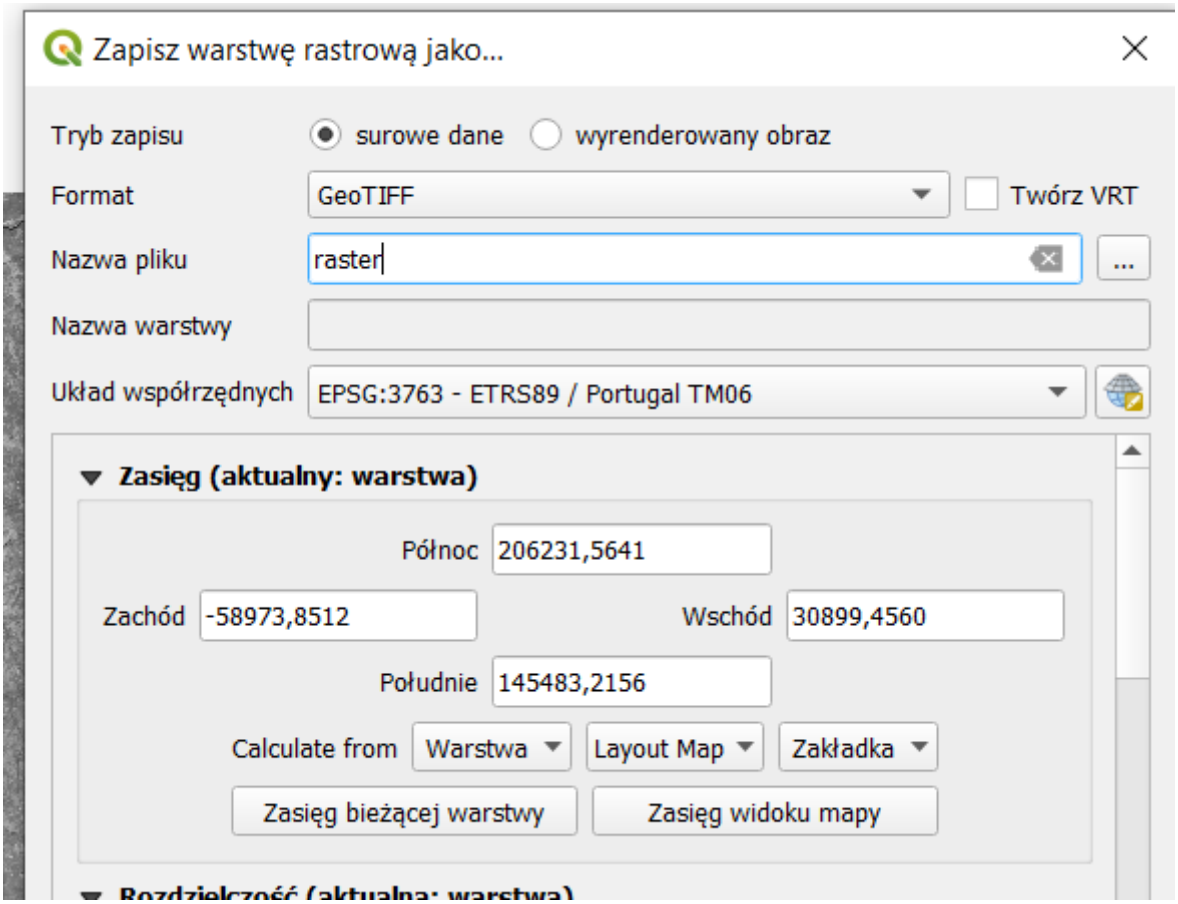
porto_ndvi2 | Enter a SQL expression to filter results (use Ctrl+Space)

Grid | Text

	rid	rast
1	245	01000001003849EE0BB84F3E404F2001E9AEB43DC0086F60B09B56E3C01F1195122C32044
2	246	01000001003849EE0BB84F3E404F2001E9AEB43DC03EF15EB9D152E3C075C249A2725504
3	270	01000001003849EE0BB84F3E404F2001E9AEB43DC075735DC2074FE3C0E4374673AE1104
4	246	01000001003849EE0BB84F3E404F2001E9AEB43DC075B133BEF0E4E2C00224CA124B7E04
5	247	01000001003849EE0BB84F3E404F2001E9AEB43DC0AB0CA038D66DE1C079E13D89743B0
6	270	01000001003849EE0BB84F3E404F2001E9AEB43DC03E4E2033AFB3E2C0E4374673AE1104
7	221	01000001003849EE0BB84F3E404F2001E9AEB43DC0C069108BB315E5C0B6C41B00DC9004
8	244	01000001003849EE0BB84F3E404F2001E9AEB43DC08A8A5008A0B8E5C02644A7A801820
9	245	01000001003849EE0BB84F3E404F2001E9AEB43DC0D2D51D3ACD37E5C0653CEA2E8188(
10	244	01000001003849EE0BB84F3E404F2001E9AEB43DC0D4DDC44A29E0E6C014B4B85D26800
11	268	01000001003849EE0BB84F3E404F2001E9AEB43DC0AF24956AEA66E6C0E4374673AE1104
12	221	01000001003849EE0BB84F3E404F2001E9AEB43DC0E31A995A96E6E3C0EC74E7E06D9604
13	222	01000001003849EE0BB84F3E404F2001E9AEB43DC03EF15EB9D152E3C0EC74E7E06D9604
14	245	01000001003849EE0BB84F3E404F2001E9AEB43DC08863BE794F45E4C0653CEA2E818804
15	246	01000001003849EE0BB84F3E404F2001E9AEB43DC03EF15EB9D152E3C0653CEA2E818804
16	245	01000001003849EE0BB84F3E404F2001E9AEB43DC09BB0E0AA749CE4C07E0AC14760560
17	246	01000001003849EE0BB84F3E404F2001E9AEB43DC03EF15EB9D152E3C0C121F8B4E14204
18	269	01000001003849EE0BB84F3E404F2001E9AEB43DC008F3B3B8C92AE4C0E4374673AE1104
19	270	01000001003849EE0BB84F3E404F2001E9AEB43DC03EF15EB9D152E3C0E4374673AE1104
20	244	01000001003849EE0BB84F3E404F2001E9AEB43DC09CF65DB7B9DAE5C0001A3B3B61490
21	245	01000001003849EE0BB84F3E404F2001E9AEB43DC0D2D51D3ACD37E5C0E541D54A9846(
22	268	01000001003849EE0BB84F3E404F2001E9AEB43DC02E114C2342ADE5C0E4374673AE1104
23	269	01000001003849EE0BB84F3E404F2001E9AEB43DC0D2D51D3ACD37E5C0E4374673AE11(

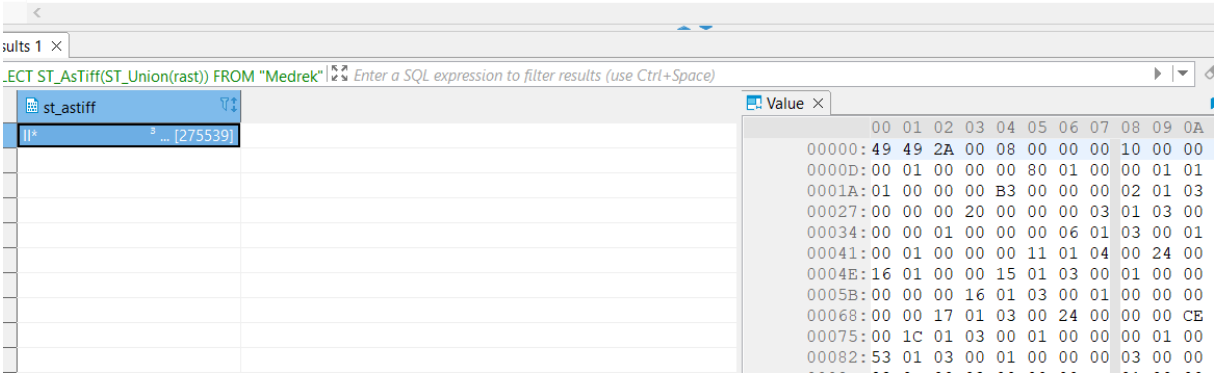
Eksport danych

Przykład 0



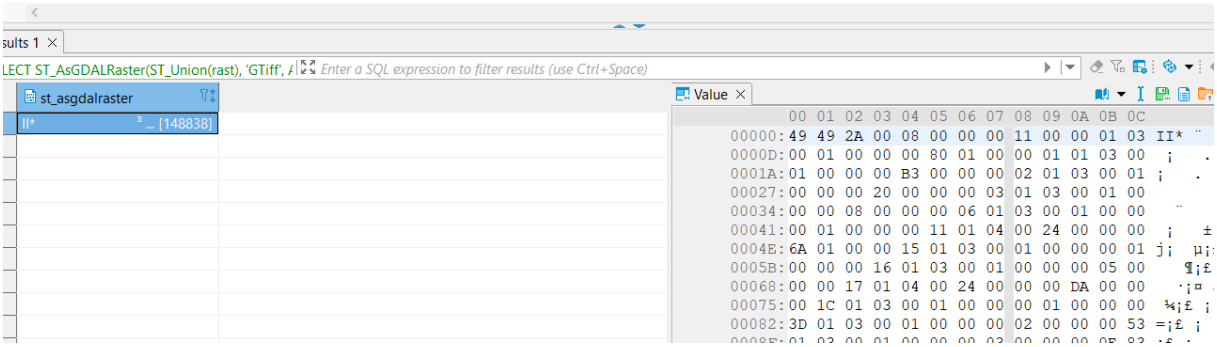
Przykład 1

```
--Przykład 1

SELECT ST_AsTiff(ST_Union(rast))
FROM "Medrek".porto_ndvi;
```

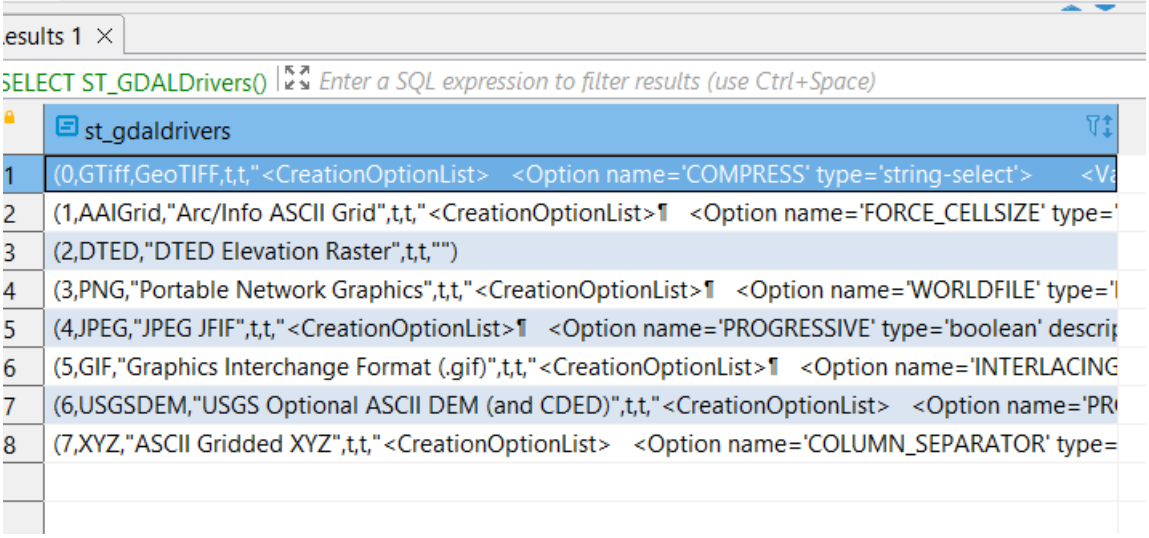


Przykład 2

```
--PRZYKŁAD 2
SELECT ST_AsGDALRaster(ST_Union(rast), 'GTiff', ARRAY['COMPRESS=DEFLATE',
'PREDICTOR=2', 'PZLEVEL=9'])
FROM "Medrek".porto_ndvi;
```




```
--Przykład 3
SELECT ST_GDALDrivers();
```



Przykład 3

```
--Przykład 3

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 "Medrek".porto_ndvi;
----------------------------------------------
SELECT lo_export(loid, 'D:\raster.tiff')
 FROM tmp_out;
----------------------------------------------
SELECT lo_unlink(loid)
 FROM tmp_out;
```