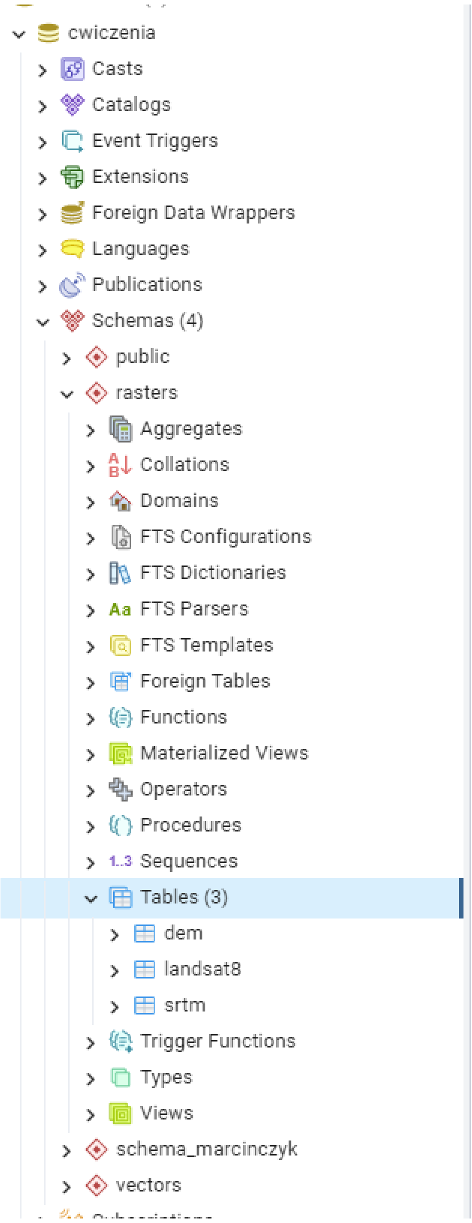
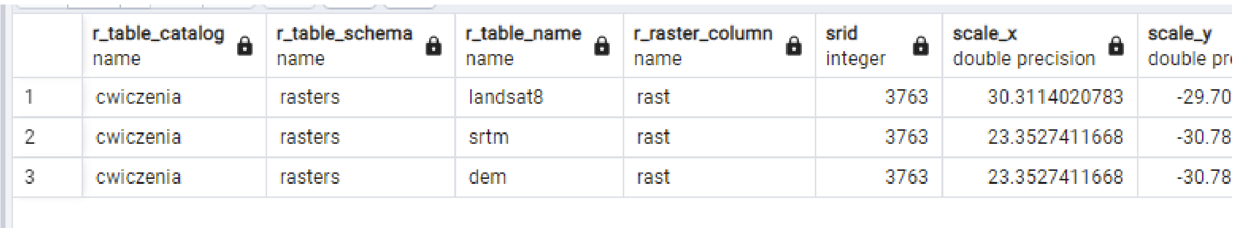
Załadowanie potrzebnych danych

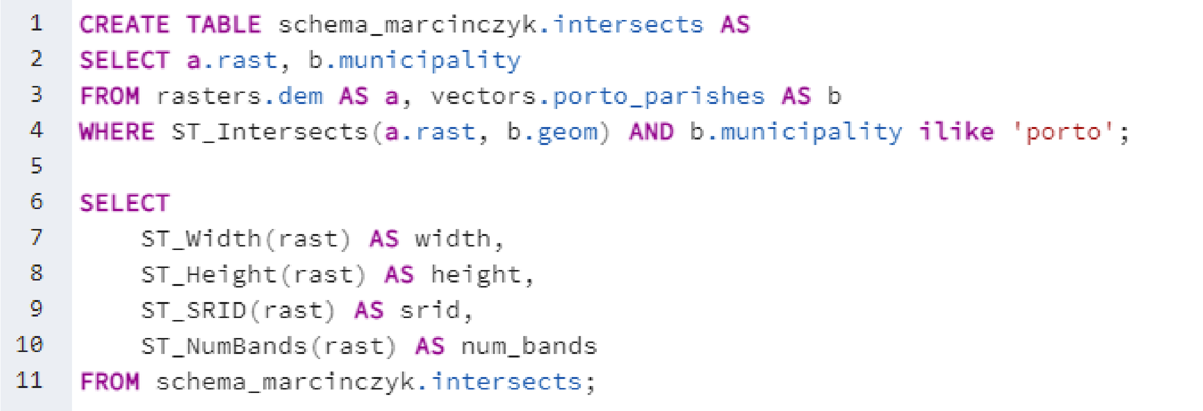


- cwiczenia
  - Casts
  - Catalogs
  - Event Triggers
  - Extensions
  - Foreign Data Wrappers
  - Languages
  - Publications
  - Schemas (4)
    - public
    - rasters
      - Aggregates
      - Collations
      - Domains
      - FTS Configurations
      - FTS Dictionaries
      - FTS Parsers
      - FTS Templates
      - Foreign Tables
      - Functions
      - Materialized Views
      - Operators
      - Procedures
      - Sequences
      - Tables (3)
        - dem
        - landsat8
        - srtm
      - Trigger Functions
      - Types
      - Views
    - schema_marcinczyk
    - vectors

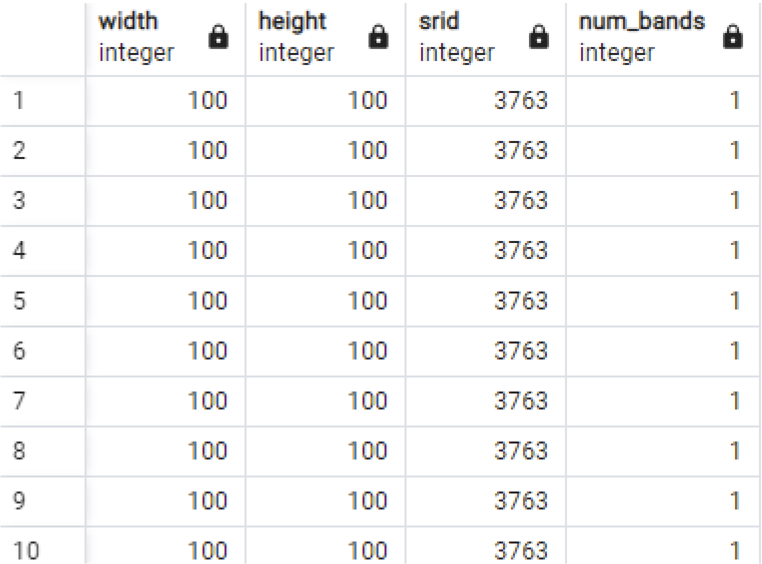


	r_table_catalog name	r_table_schema name	r_table_name name	r_raster_column name	srid integer	scale_x double precision	scale_y double pr
1	cwiczenia	rasters	landsat8	rast	3763	30.3114020783	-29.70
2	cwiczenia	rasters	srtm	rast	3763	23.3527411668	-30.78
3	cwiczenia	rasters	dem	rast	3763	23.3527411668	-30.78

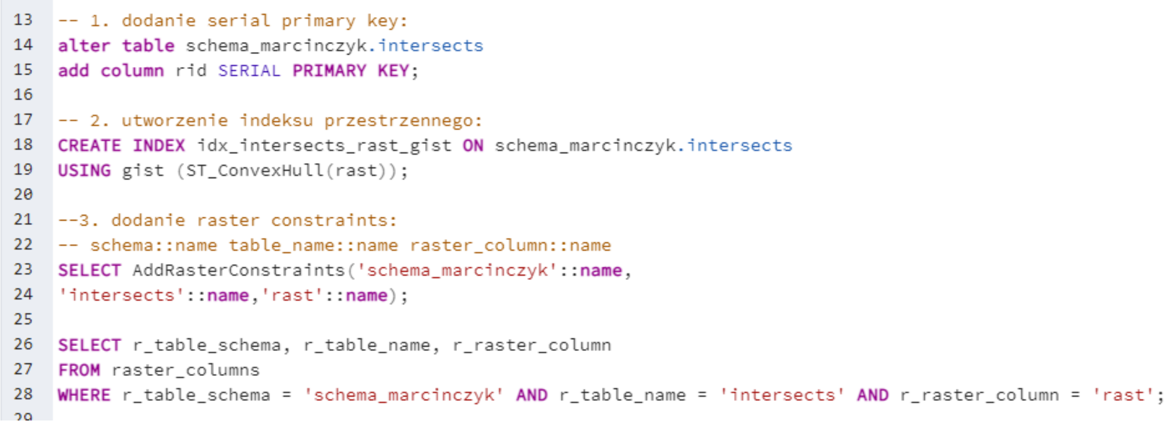
Przykład 1 - ST_Intersects



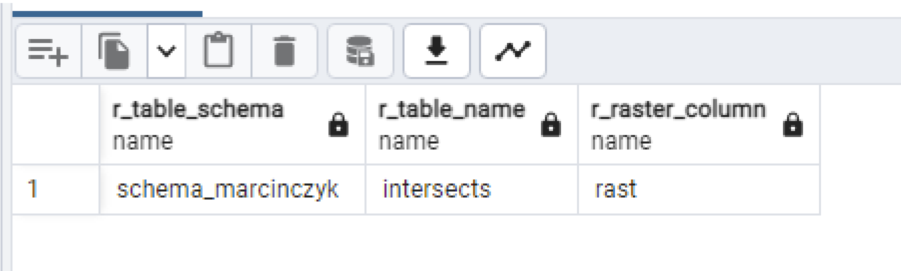
```sql
CREATE TABLE schema_marcinczyk.intersects AS
SELECT a.rast, b.municipality
FROM rasters.dem AS a, vectors.porto_parishes AS b
WHERE ST_Intersects(a.rast, b.geom) AND b.municipality ilike 'porto';

SELECT
    ST_Width(rast) AS width,
    ST_Height(rast) AS height,
    ST_SRID(rast) AS srid,
    ST_NumBands(rast) AS num_bands
FROM schema_marcinczyk.intersects;
```



	width integer	height integer	srid integer	num_bands integer
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

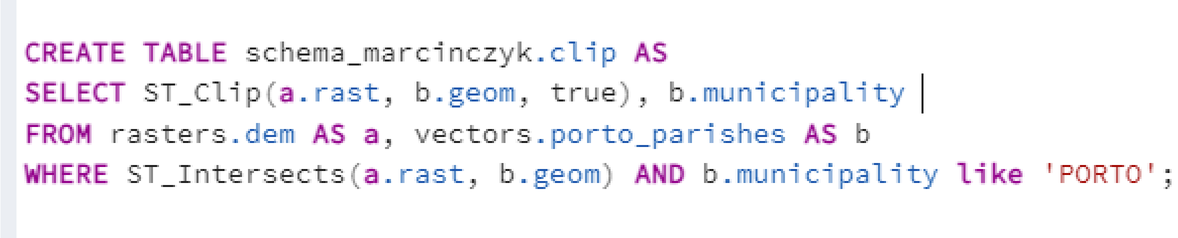


```sql
-- 1. dodanie serial primary key:
alter table schema_marcinczyk.intersects
add column rid SERIAL PRIMARY KEY;

-- 2. utworzenie indeksu przestrzennego:
CREATE INDEX idx_intersects_rast_gist ON schema_marcinczyk.intersects
USING gist (ST_ConvexHull(rast));

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

SELECT r_table_schema, r_table_name, r_raster_column
FROM raster_columns
WHERE r_table_schema = 'schema_marcinczyk' AND r_table_name = 'intersects' AND r_raster_column = 'rast';
```

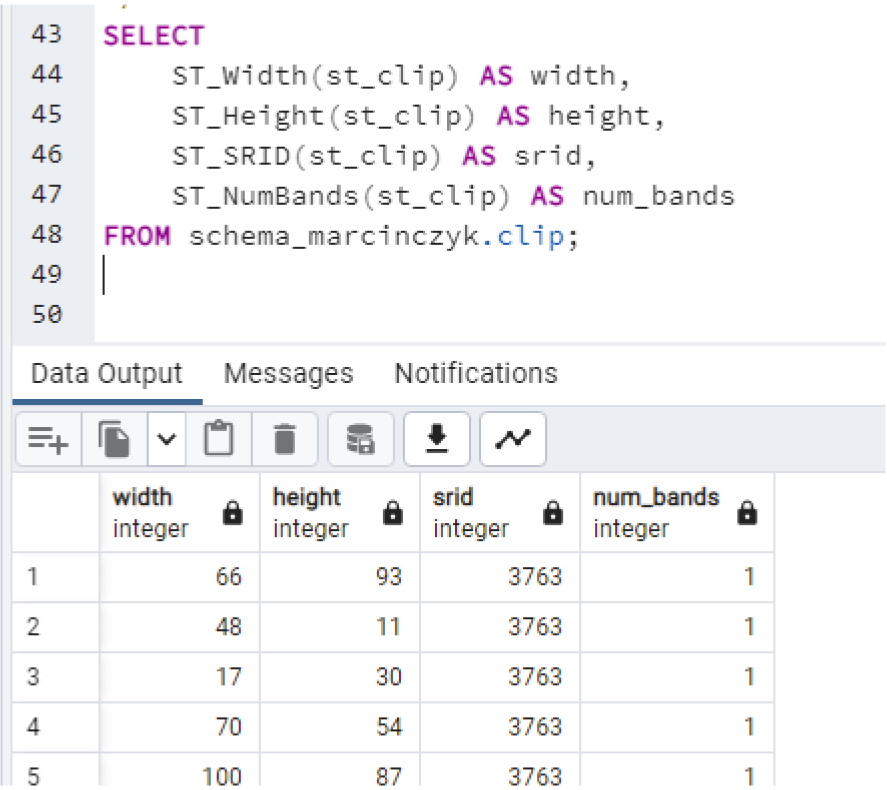


	r_table_schema name	r_table_name name	r_raster_column name
1	schema_marcinczyk	intersects	rast

Przykład 2 - ST_Clip



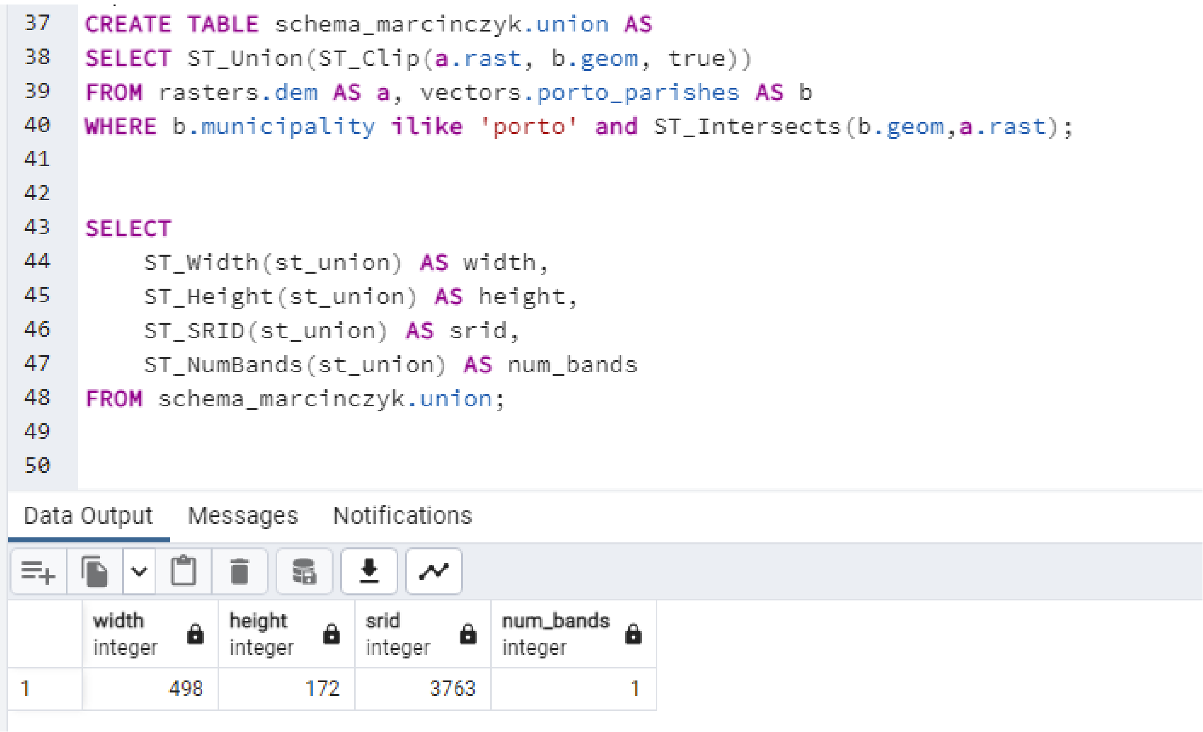
```
CREATE TABLE schema_marcinczyk.clip AS
SELECT ST_Clip(a.rast, b.geom, true), b.municipality
FROM rasters.dem AS a, vectors.porto_parishes AS b
WHERE ST_Intersects(a.rast, b.geom) AND b.municipality like 'PORTO';
```



```
SELECT
    ST_Width(st_clip) AS width,
    ST_Height(st_clip) AS height,
    ST_SRID(st_clip) AS srid,
    ST_NumBands(st_clip) AS num_bands
FROM schema_marcinczyk.clip;
```

	width integer	height integer	srid integer	num_bands integer
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

Przykład 3 - ST_Union

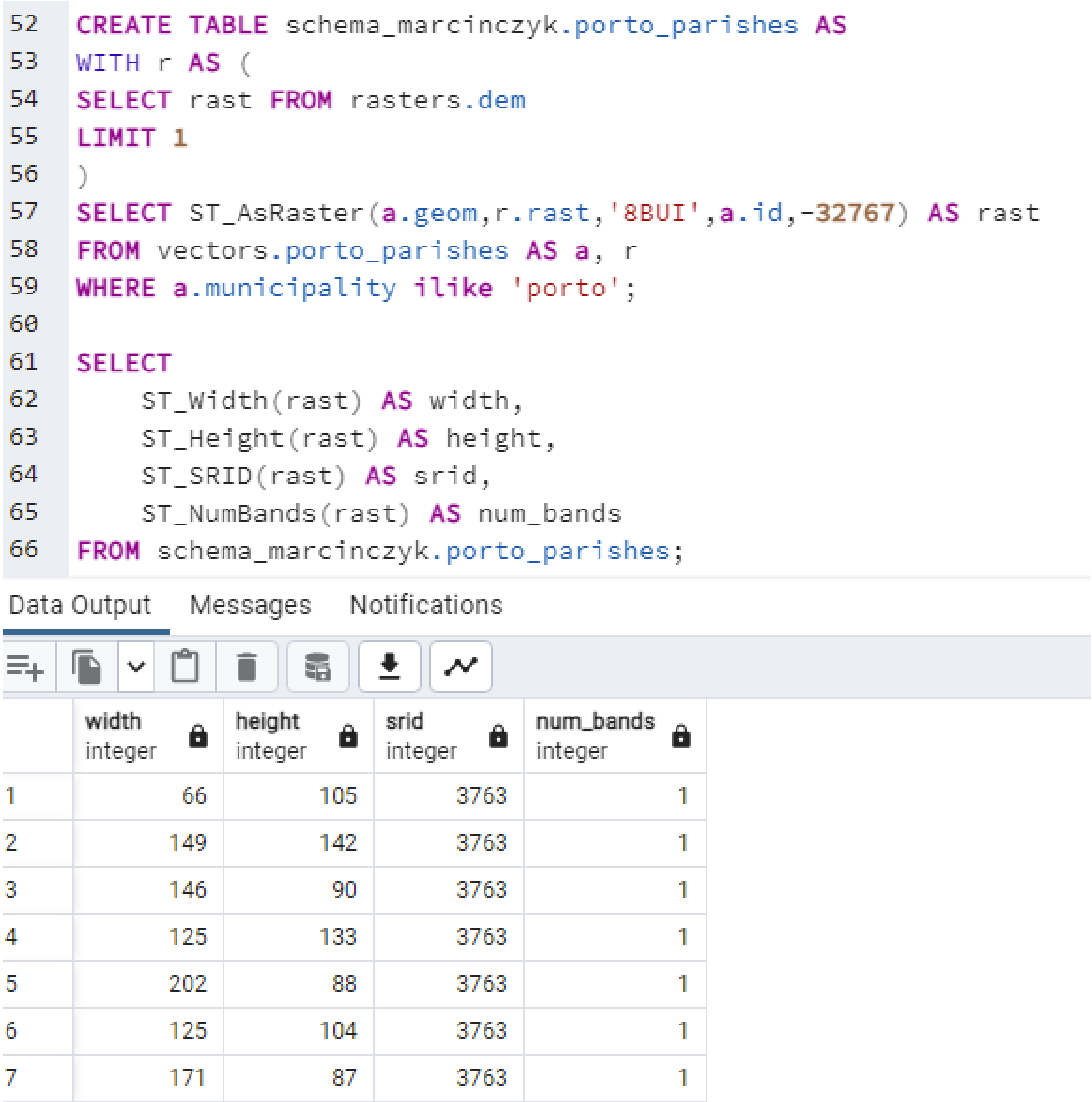


```sql
CREATE TABLE schema_marcinczyk.union AS
SELECT ST_Union(ST_Clip(a.rast, b.geom, true))
FROM rasters.dem AS a, vectors.porto_parishes AS b
WHERE b.municipality ilike 'porto' and ST_Intersects(b.geom,a.rast);


SELECT
    ST_Width(st_union) AS width,
    ST_Height(st_union) AS height,
    ST_SRID(st_union) AS srid,
    ST_NumBands(st_union) AS num_bands
FROM schema_marcinczyk.union;
```

	width integer	height integer	srid integer	num_bands integer
1	498	172	3763	1

Tworzenie rastrów z wektorów

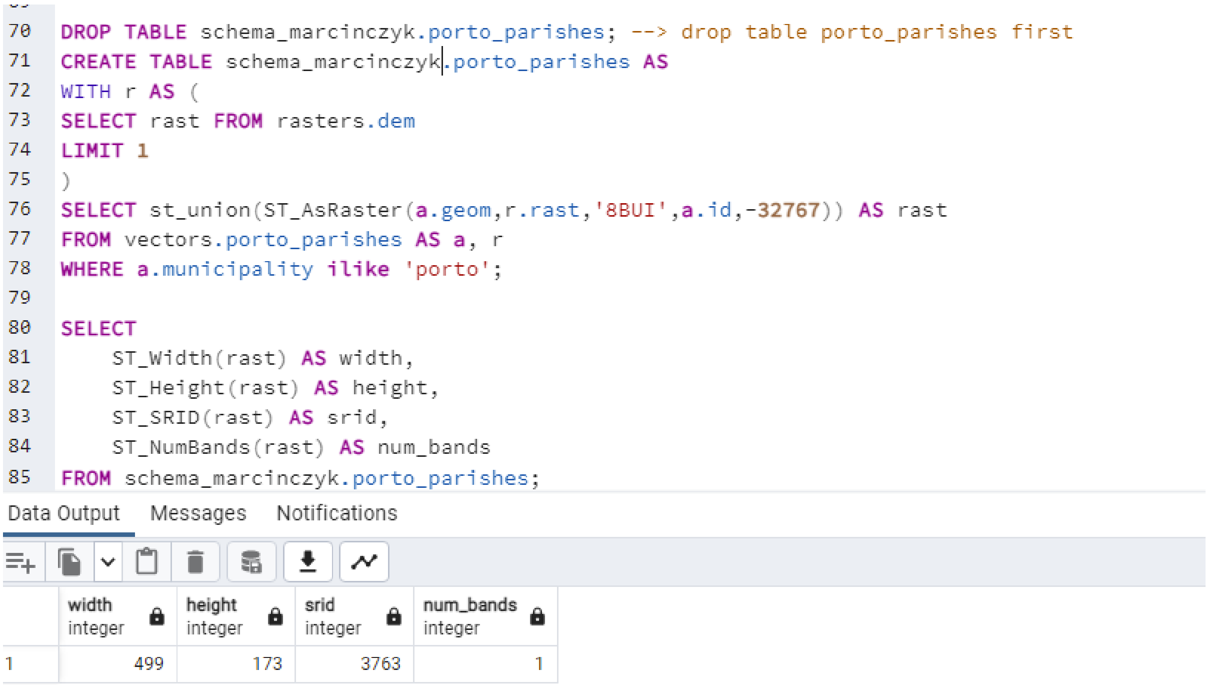
Przykład 1 - ST_AsRaster



```sql
CREATE TABLE schema_marcinczyk.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';

SELECT
    ST_Width(rast) AS width,
    ST_Height(rast) AS height,
    ST_SRID(rast) AS srid,
    ST_NumBands(rast) AS num_bands
FROM schema_marcinczyk.porto_parishes;
```

	width integer	height integer	srid integer	num_bands integer
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

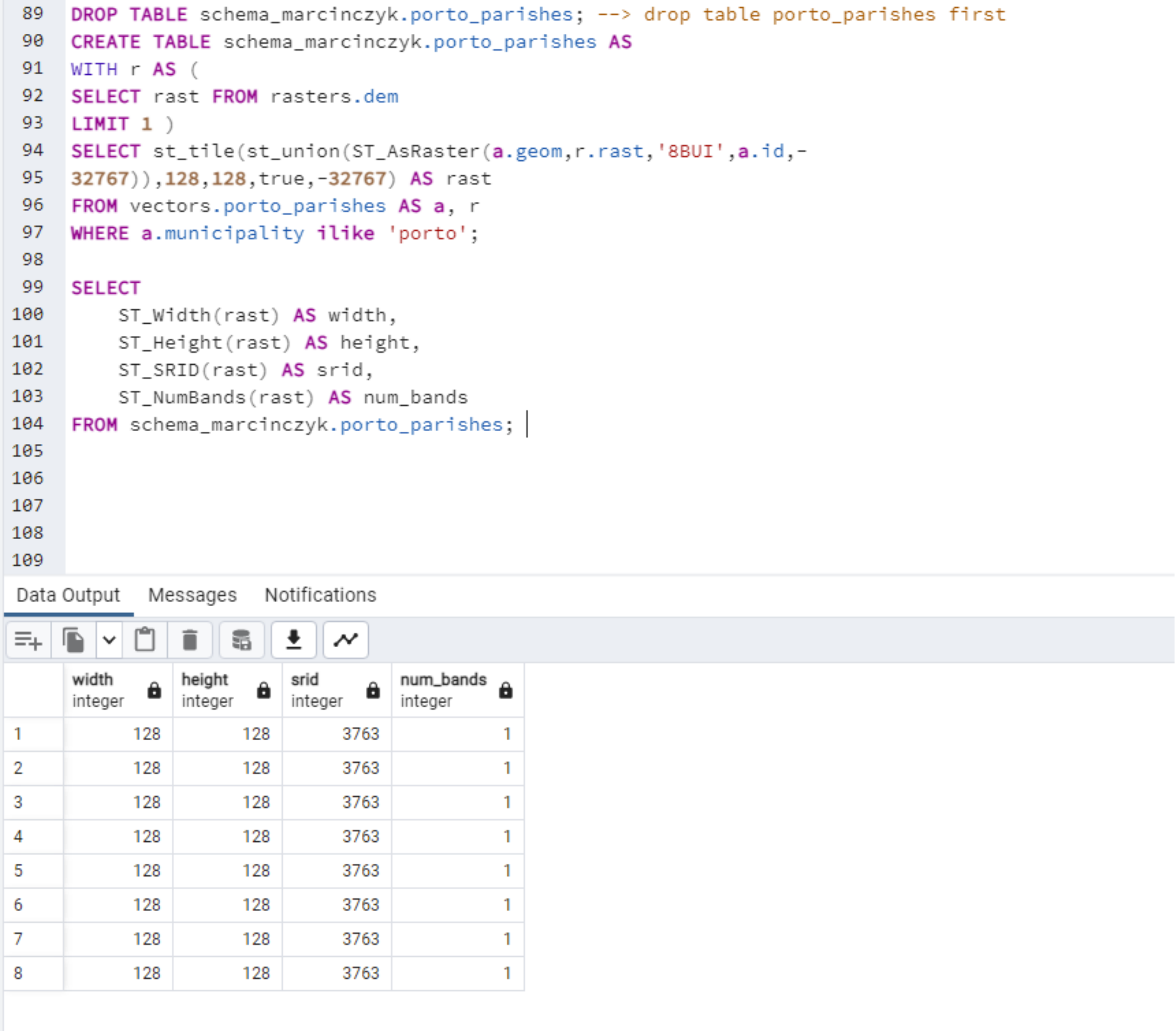
Przykład 2 - ST_Union



```
DROP TABLE schema_marcinczyk.porto_parishes; --> drop table porto_parishes first
CREATE TABLE schema_marcinczyk.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';

SELECT
    ST_Width(rast) AS width,
    ST_Height(rast) AS height,
    ST_SRID(rast) AS srid,
    ST_NumBands(rast) AS num_bands
FROM schema_marcinczyk.porto_parishes;
```

	width integer	height integer	srid integer	num_bands integer
1	499	173	3763	1

Przykład 3 - ST_Tile

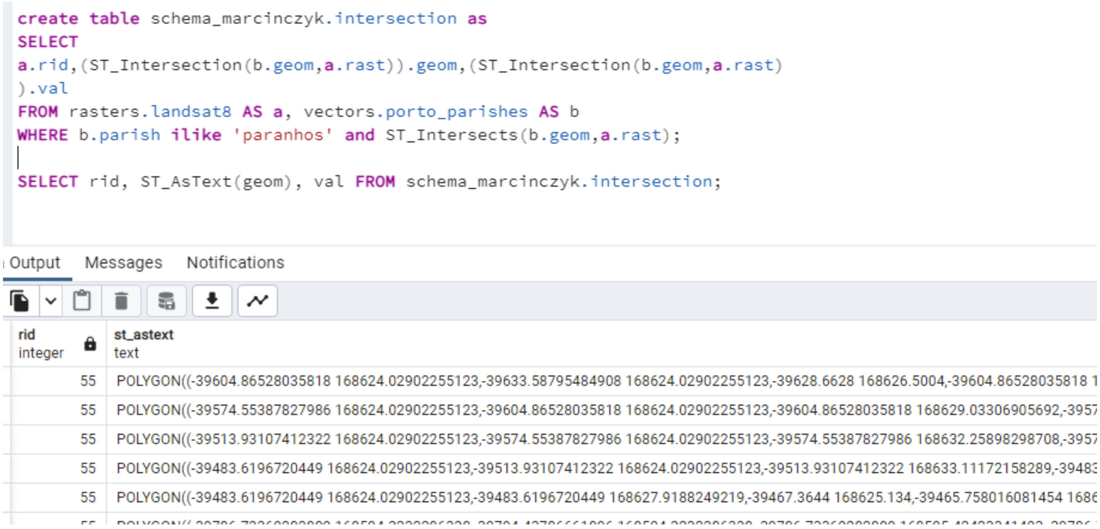


```
DROP TABLE schema_marcinczyk.porto_parishes; --> drop table porto_parishes first
CREATE TABLE schema_marcinczyk.porto_parishes AS
WITH r AS (
SELECT rast FROM rasters.dem
LIMIT 1 )
SELECT st_tile(st_union(ST_AsRaster(a.geom,r.rast,'8BUI',a.id,-
32767)),128,128,true,-32767) AS rast
FROM vectors.porto_parishes AS a, r
WHERE a.municipality ilike 'porto';

SELECT
    ST_Width(rast) AS width,
    ST_Height(rast) AS height,
    ST_SRID(rast) AS srid,
    ST_NumBands(rast) AS num_bands
FROM schema_marcinczyk.porto_parishes;
```

	width integer	height integer	srid integer	num_bands integer
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	128	128	3763	1

Konwertowanie rastrów na wektory (wektoryzowanie)

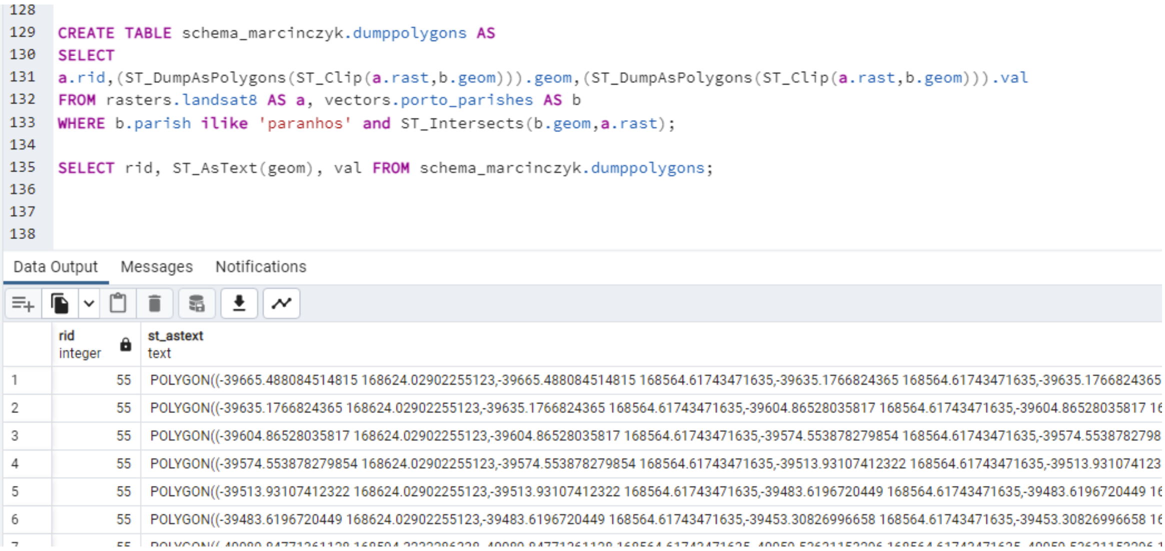
Przykład 1 - ST_Intersection



```
create table schema_marcinczyk.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);

SELECT rid, ST_AsText(geom), val FROM schema_marcinczyk.intersection;
```

rid integer	st_astext text
55	POLYGON((-39604.86528035818 168624.02902255123,-39633.58795484908 168624.02902255123,-39628.6628 168626.5004,-39604.86528035818 1
55	POLYGON((-39574.55387827986 168624.02902255123,-39604.86528035818 168624.02902255123,-39604.86528035818 168629.03306905692,-3957
55	POLYGON((-39513.93107412322 168624.02902255123,-39574.55387827986 168624.02902255123,-39574.55387827986 168632.25898298708,-3957
55	POLYGON((-39483.6196720449 168624.02902255123,-39513.93107412322 168624.02902255123,-39513.93107412322 168633.11172158289,-39483
55	POLYGON((-39483.6196720449 168624.02902255123,-39483.6196720449 168627.9188249219,-39467.3644 168625.134,-39465.758016081454 1686

Przykład 2 - ST_DumpAsPolygons

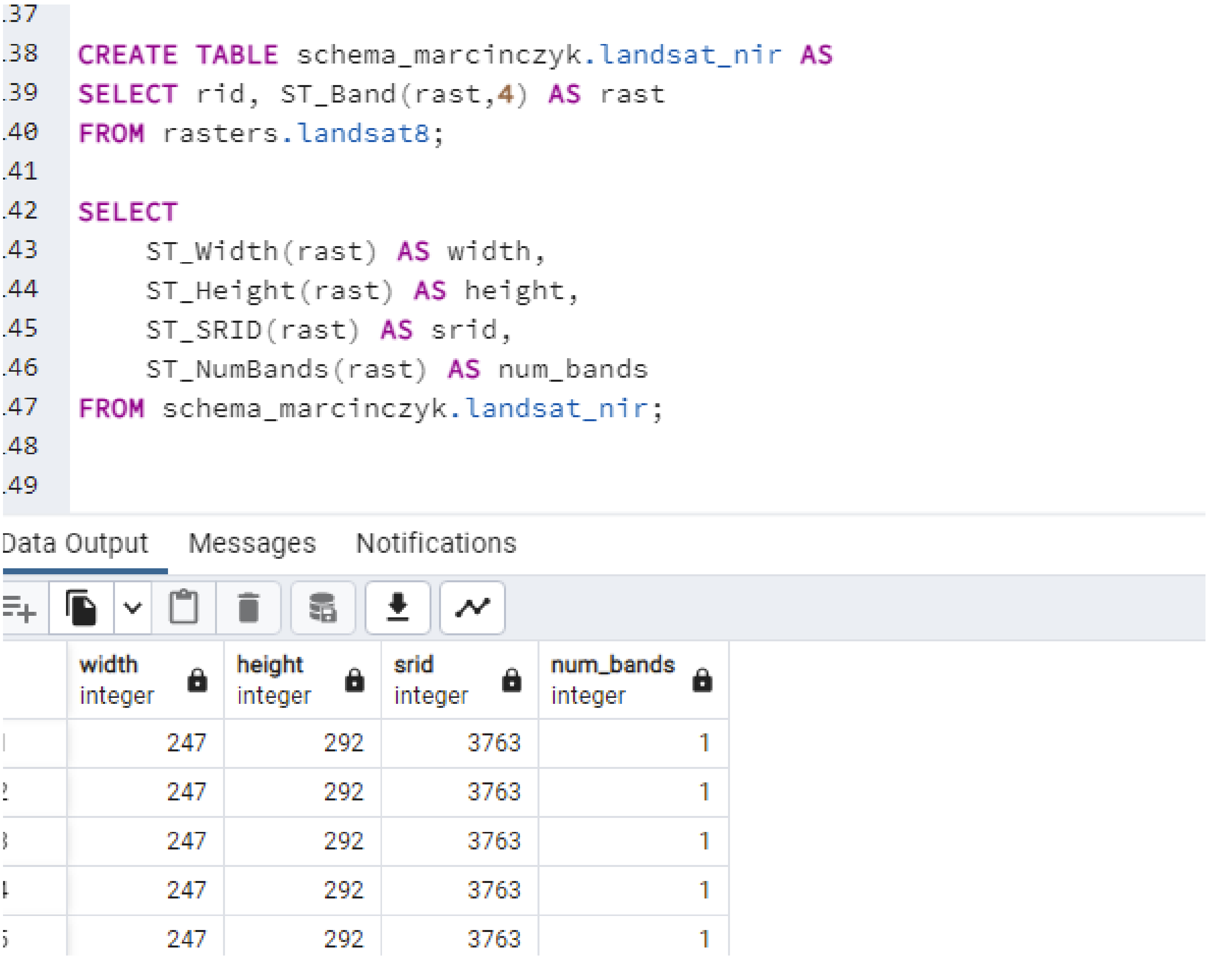


```
CREATE TABLE schema_marcinczyk.dumppolygons AS
SELECT
a.rid,(ST_DumpAsPolygons(ST_Clip(a.rast,b.geom))).geom,(ST_DumpAsPolygons(ST_Clip(a.rast,b.geom))).val
FROM rasters.landsat8 AS a, vectors.porto_parishes AS b
WHERE b.parish ilike 'paranhos' and ST_Intersects(b.geom,a.rast);

SELECT rid, ST_AsText(geom), val FROM schema_marcinczyk.dumppolygons;
```

	rid integer	st_astext text
1	55	POLYGON((-39665.488084514815 168624.02902255123,-39665.488084514815 168564.61743471635,-39635.1766824365 168564.61743471635,-39635.1766824365
2	55	POLYGON((-39635.1766824365 168624.02902255123,-39635.1766824365 168564.61743471635,-39604.86528035817 168564.61743471635,-39604.86528035817 16
3	55	POLYGON((-39604.86528035817 168624.02902255123,-39604.86528035817 168564.61743471635,-39574.553878279854 168564.61743471635,-39574.5538782798
4	55	POLYGON((-39574.553878279854 168624.02902255123,-39574.553878279854 168564.61743471635,-39513.93107412322 168564.61743471635,-39513.931074123
5	55	POLYGON((-39513.93107412322 168624.02902255123,-39513.93107412322 168564.61743471635,-39483.6196720449 168564.61743471635,-39483.6196720449 16
6	55	POLYGON((-39483.6196720449 168624.02902255123,-39483.6196720449 168564.61743471635,-39453.30826996658 168564.61743471635,-39453.30826996658 16

Analiza Rastrów

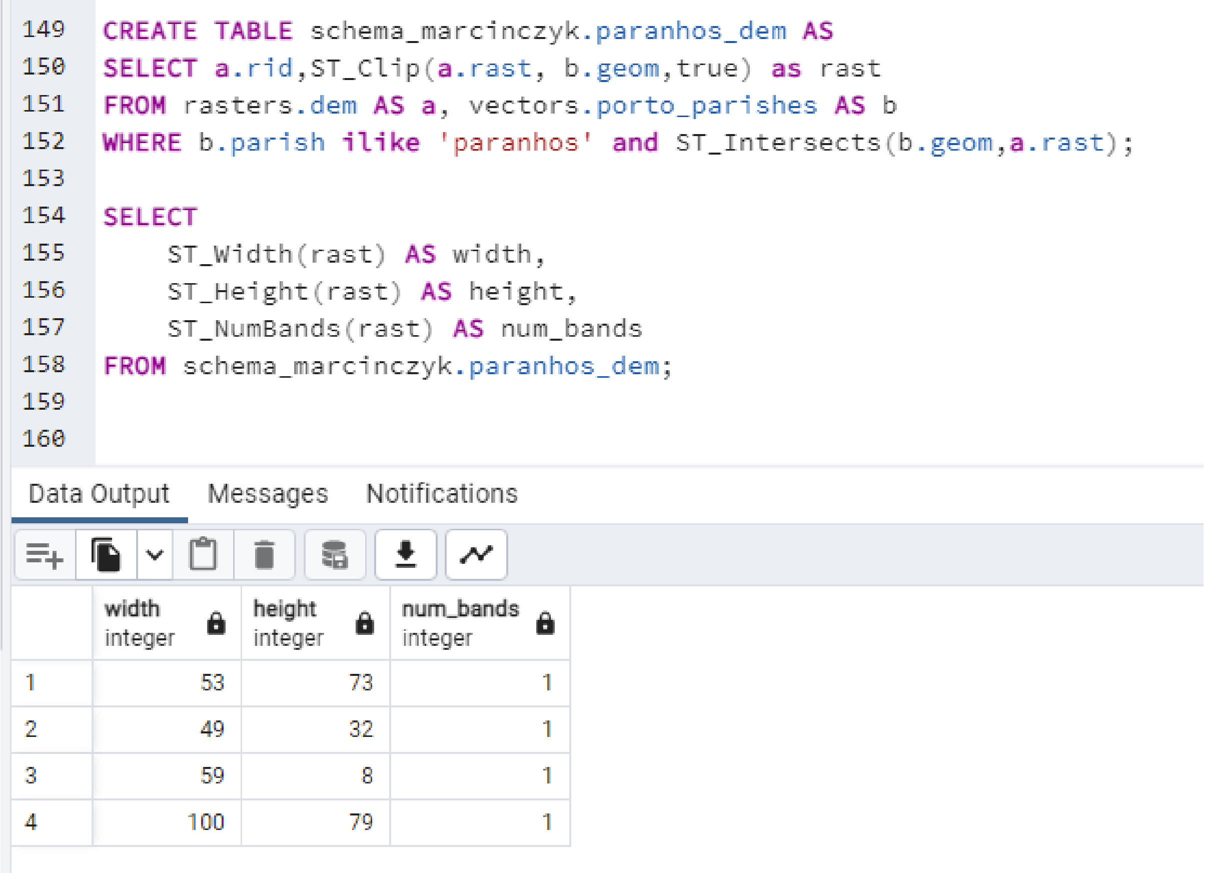
Przykład 1 - ST_Band



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

SELECT
    ST_Width(rast) AS width,
    ST_Height(rast) AS height,
    ST_SRID(rast) AS srid,
    ST_NumBands(rast) AS num_bands
FROM schema_marcinczyk.landsat_nir;
```

	width integer	height integer	srid integer	num_bands integer
1	247	292	3763	1
2	247	292	3763	1
3	247	292	3763	1
4	247	292	3763	1
5	247	292	3763	1

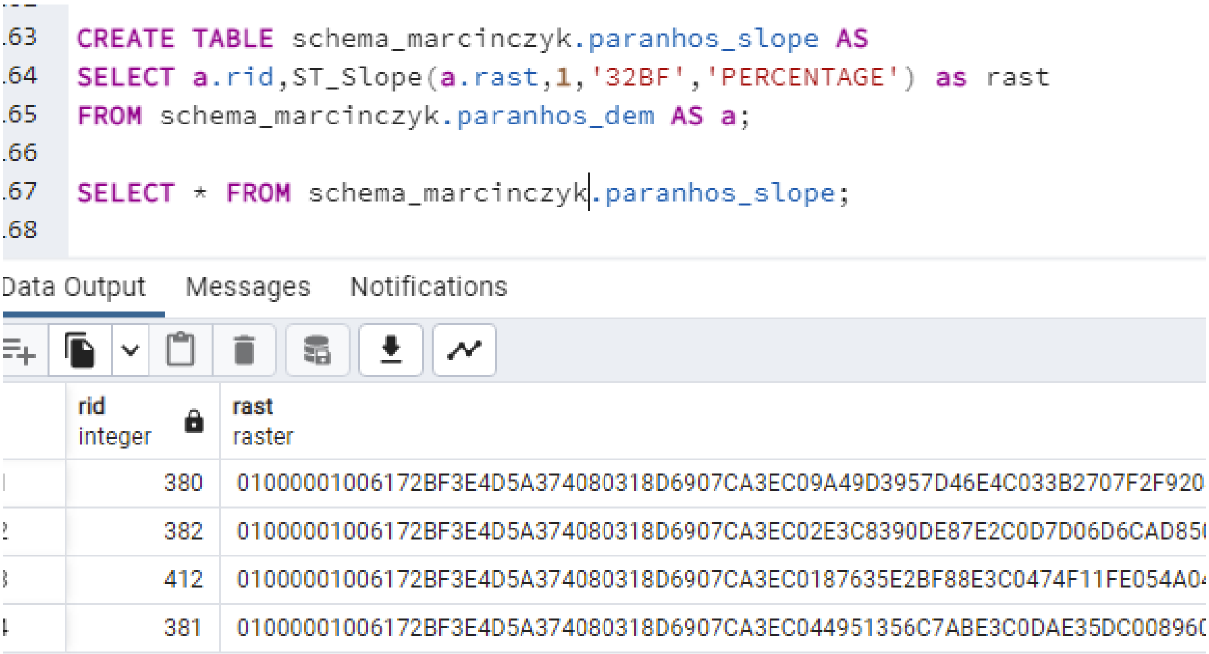
Przykład 2 - ST_Clip



```sql
CREATE TABLE schema_marcinczyk.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);

SELECT
    ST_Width(rast) AS width,
    ST_Height(rast) AS height,
    ST_NumBands(rast) AS num_bands
FROM schema_marcinczyk.paranhos_dem;
```

	width integer	height integer	num_bands integer
1	53	73	1
2	49	32	1
3	59	8	1
4	100	79	1

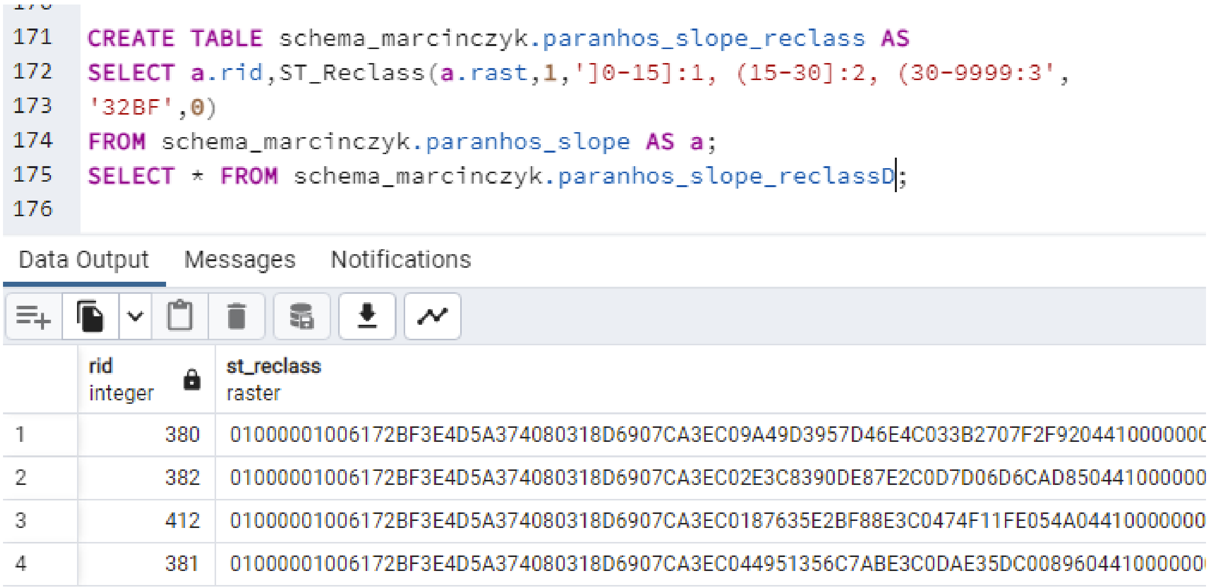
Przykład 3 - ST_Slope



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

SELECT * FROM schema_marcinczyk.paranhos_slope;
```

	rid integer	rast raster
1	380	0100000100617 2BF3E4D5A374080318D6907CA3EC09A49D3957D46E4C033B2707F2F920
2	382	01000001006172BF3E4D5A374080318D6907CA3EC02E3C8390DE87E2C0D7D06D6CAD85
3	412	01000001006172BF3E4D5A374080318D6907CA3EC0187635E2BF88E3C0474F11FE054A0
4	381	01000001006172BF3E4D5A374080318D6907CA3EC044951356C7ABE3C0DAE35DC00896

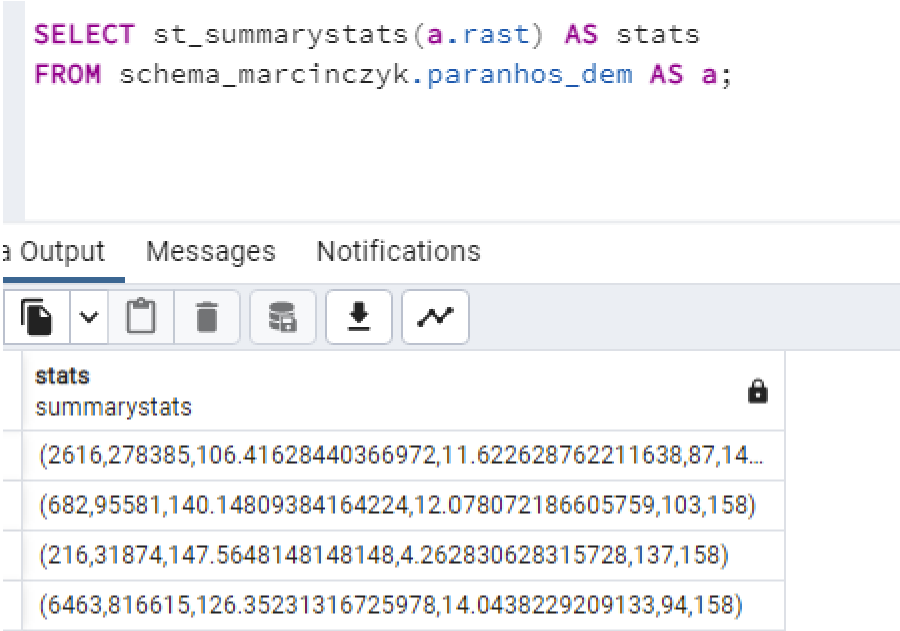
Przykład 4 - ST_Reclass



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

	rid integer	st_reclass raster
1	380	01000001006172BF3E4D5A374080318D6907CA3EC09A49D3957D46E4C033B2707F2F9204410000000
2	382	01000001006172BF3E4D5A374080318D6907CA3EC02E3C8390DE87E2C0D7D06D6CAD850441000000
3	412	01000001006172BF3E4D5A374080318D6907CA3EC0187635E2BF88E3C0474F11FE054A04410000000
4	381	01000001006172BF3E4D5A374080318D6907CA3EC044951356C7ABE3C0DAE35DC0089604410000000

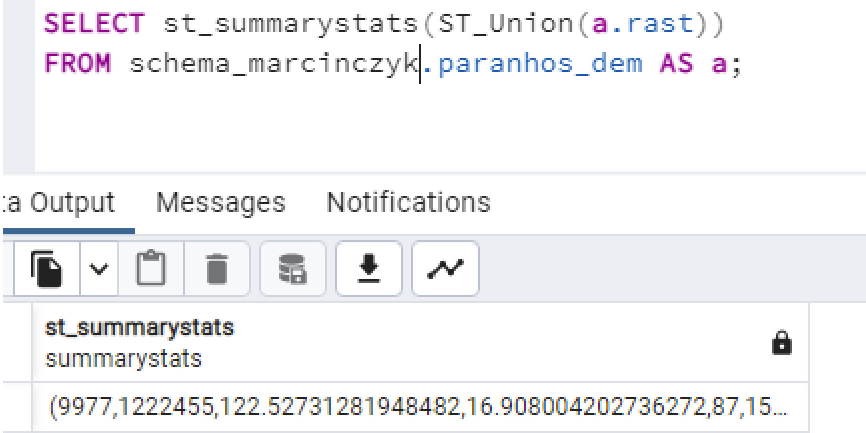
Przykład 5 - ST_SummaryStats



```sql
SELECT st_summarystats(a.rast) AS stats
FROM schema_marcinczyk.paranhos_dem AS a;
```

stats summarystats
(2616,278385,106.41628440366972,11.622628762211638,87,14...
(682,95581,140.14809384164224,12.078072186605759,103,158)
(216,31874,147.5648148148148,4.262830628315728,137,158)
(6463,816615,126.35231316725978,14.0438229209133,94,158)

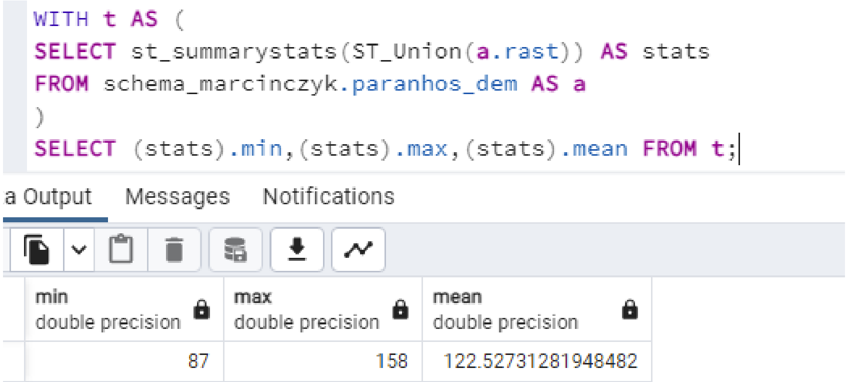
Przykład 6 - ST_SummaryStats oraz Union



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

st_summarystats summarystats
(9977,1222455,122.52731281948482,16.908004202736272,87,15...

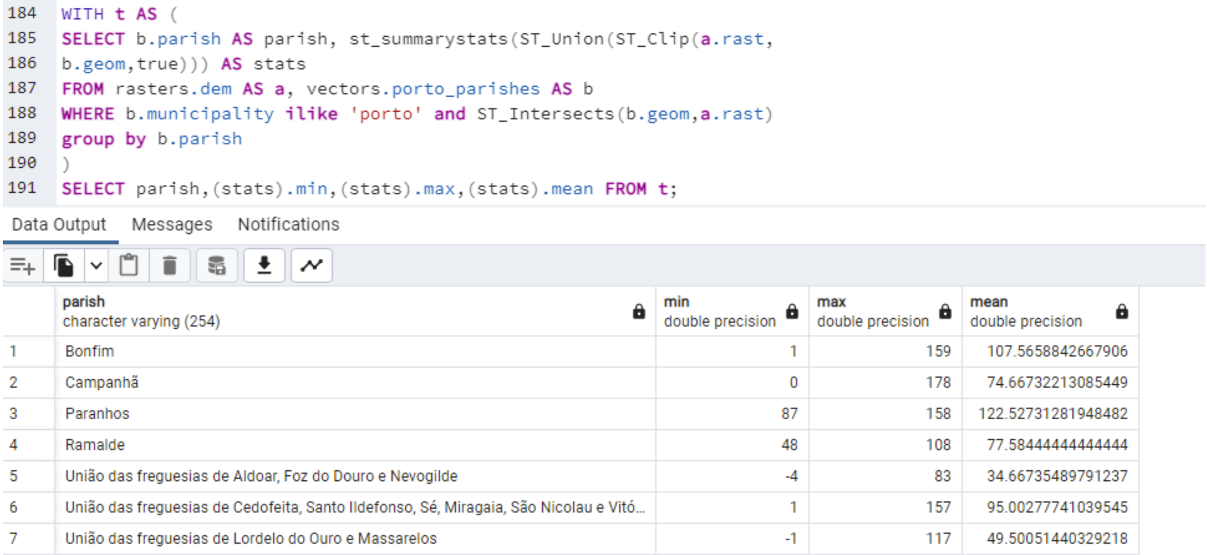
Przykład 7 - ST_SummaryStats z lepszą kontrolą złożonego typu danych



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

min double precision	max double precision	mean double precision
87	158	122.52731281948482

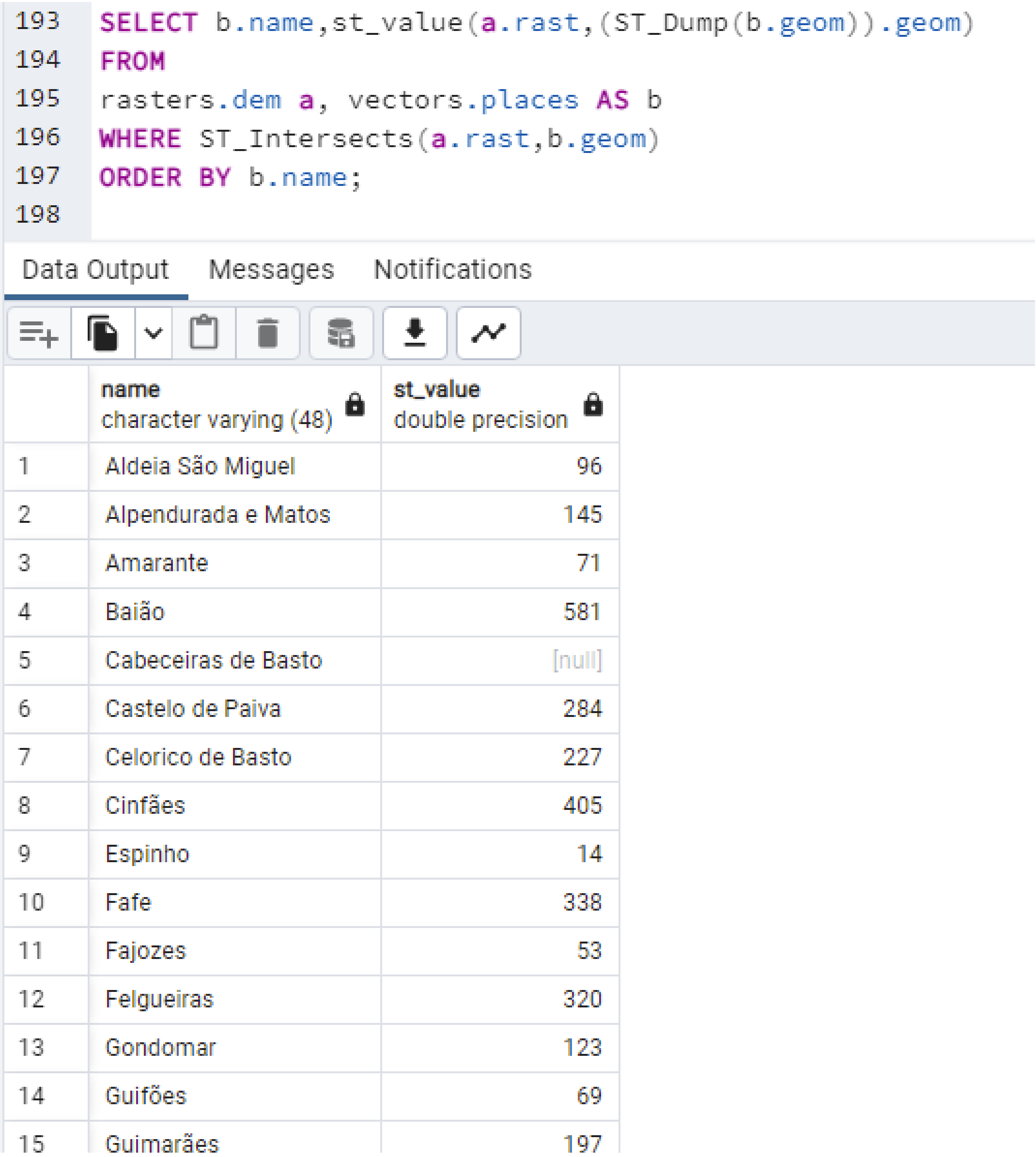
Przykład 8 - ST_SummaryStats w połączeniu z GROUP BY



```
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;
```

	parish character varying (254)	min double precision	max double precision	mean double precision
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 Vitó...	1	157	95.00277741039545
7	União das freguesias de Lordelo do Ouro e Massarelos	-1	117	49.50051440329218

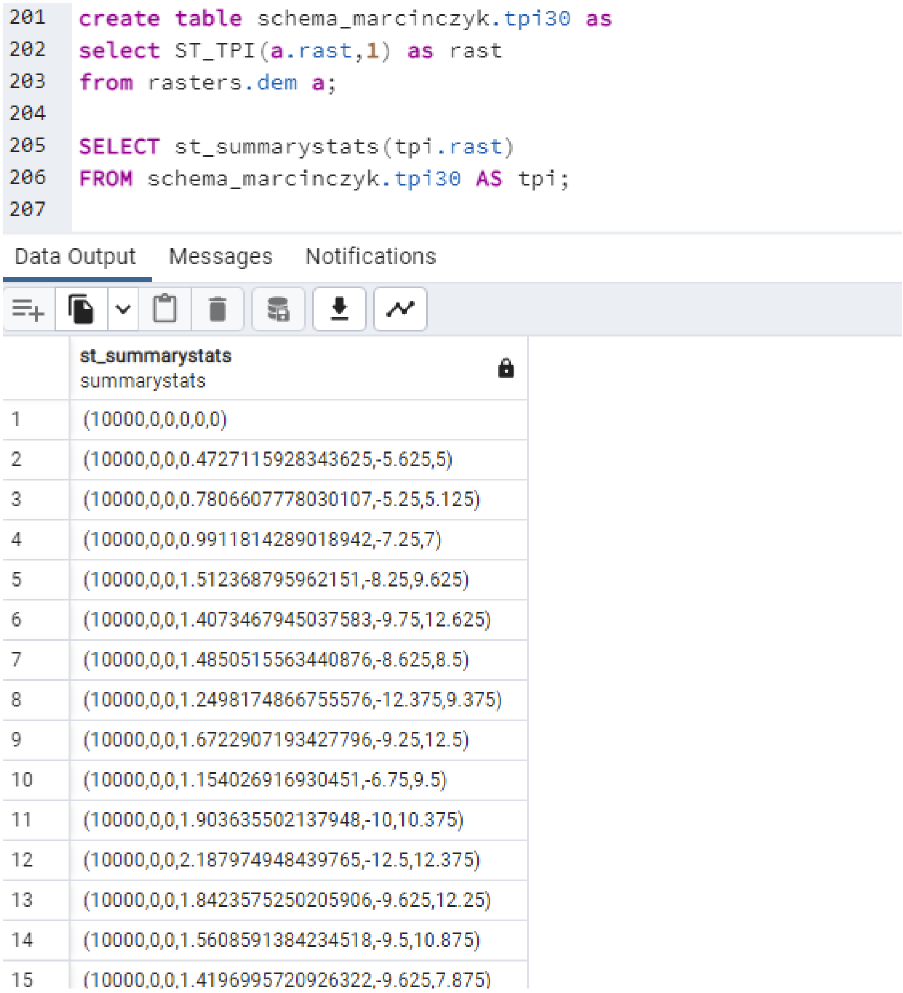
Przykład 9 - ST_Value



```sql
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;

```

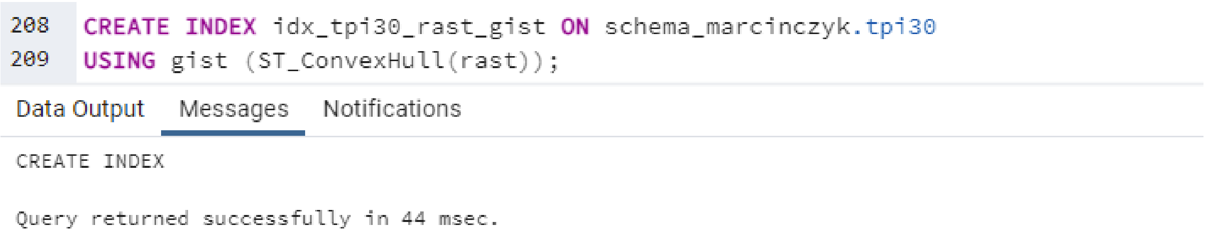
	name character varying (48)	st_value double precision
1	Aldeia São Miguel	96
2	Alpendurada e Matos	145
3	Amarante	71
4	Baião	581
5	Cabeceiras de Basto	[null]
6	Castelo de Paiva	284
7	Celorico de Basto	227
8	Cinfães	405
9	Espinho	14
10	Fafe	338
11	Fajozes	53
12	Felgueiras	320
13	Gondomar	123
14	Guifões	69
15	Guimarães	197

Przykład 10 - ST_TPI



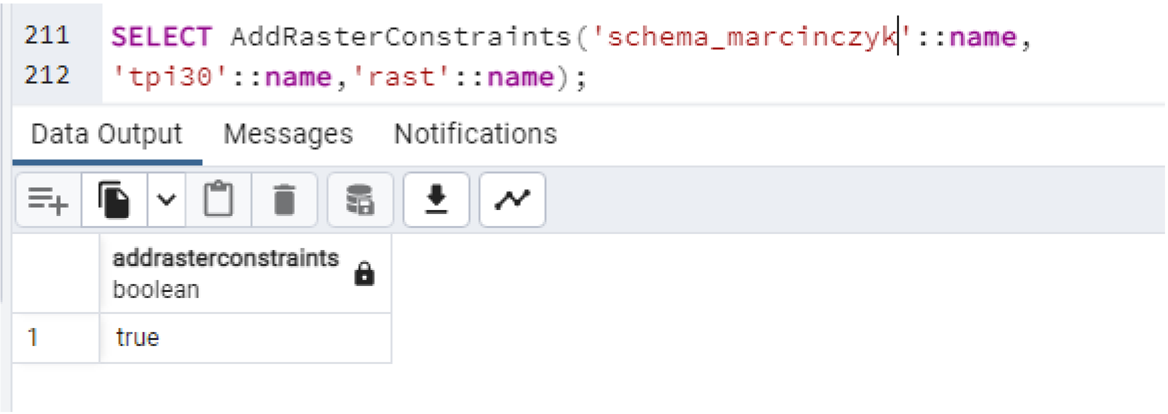
```sql
create table schema_marcinczyk.tpi30 as
select ST_TPI(a.rast,1) as rast
from rasters.dem a;

SELECT st_summarystats(tpi.rast)
FROM schema_marcinczyk.tpi30 AS tpi;
```

	st_summarystats summarystats
1	(10000,0,0,0,0,0)
2	(10000,0,0,0.4727115928343625,-5.625,5)
3	(10000,0,0,0.7806607778030107,-5.25,5.125)
4	(10000,0,0,0.9911814289018942,-7.25,7)
5	(10000,0,0,1.512368795962151,-8.25,9.625)
6	(10000,0,0,1.4073467945037583,-9.75,12.625)
7	(10000,0,0,1.4850515563440876,-8.625,8.5)
8	(10000,0,0,1.2498174866755576,-12.375,9.375)
9	(10000,0,0,1.6722907193427796,-9.25,12.5)
10	(10000,0,0,1.154026916930451,-6.75,9.5)
11	(10000,0,0,1.903635502137948,-10,10.375)
12	(10000,0,0,2.187974948439765,-12.5,12.375)
13	(10000,0,0,1.8423575250205906,-9.625,12.25)
14	(10000,0,0,1.5608591384234518,-9.5,10.875)
15	(10000,0,0,1.4196995720926322,-9.625,7.875)



```sql
CREATE INDEX idx_tpi30_rast_gist ON schema_marcinczyk.tpi30
USING gist (ST_ConvexHull(rast));
```

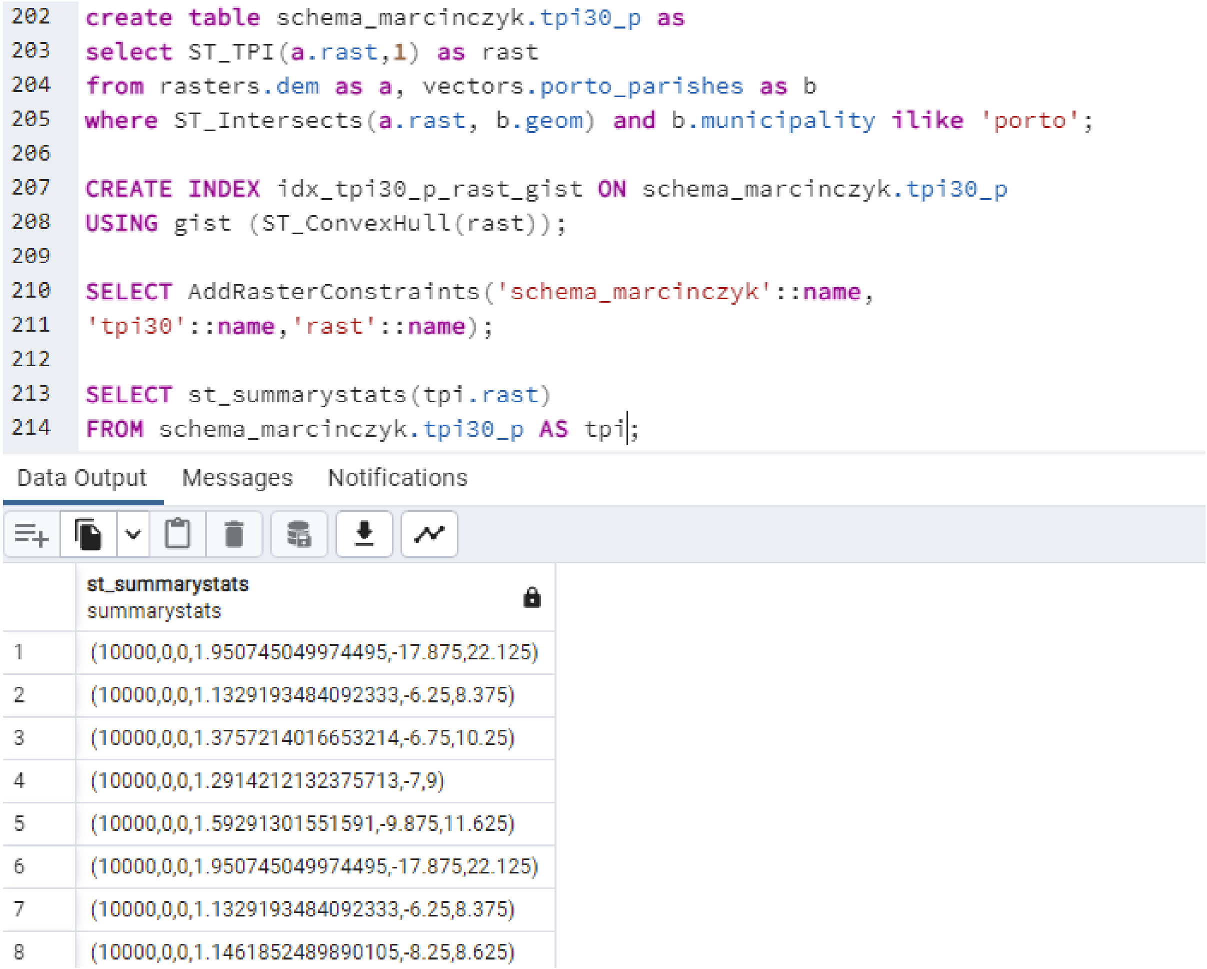
```
CREATE INDEX

Query returned successfully in 44 msec.
```



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

	addrasterconstraints boolean
1	true

Zadanie do samodzielnego rozwiązania

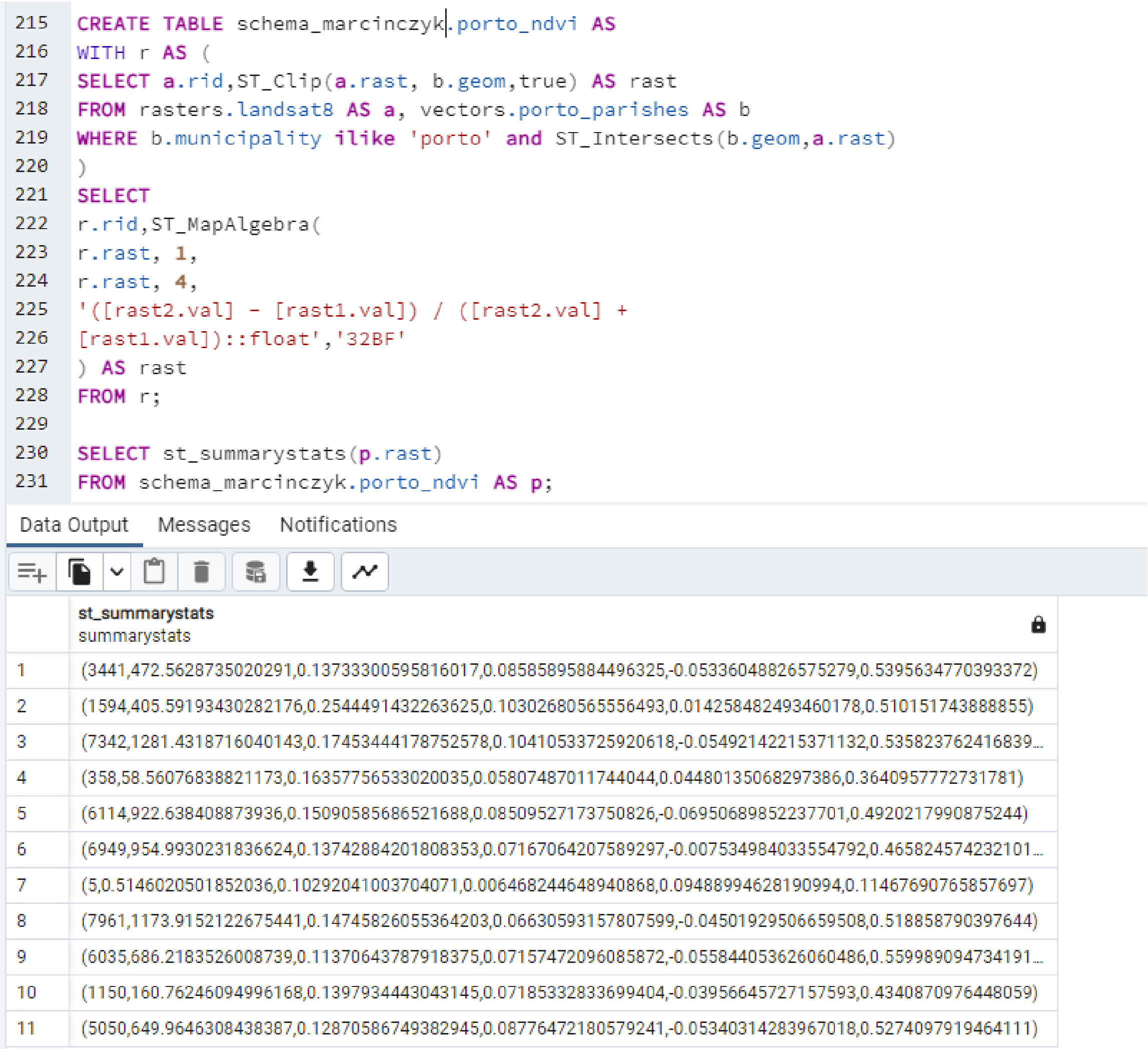


```
create table schema_marcinczyk.tpi30_p as
select ST_TPI(a.rast,1) as rast
from rasters.dem as a, vectors.porto_parishes as b
where ST_Intersects(a.rast, b.geom) and b.municipality ilike 'porto';

CREATE INDEX idx_tpi30_p_rast_gist ON schema_marcinczyk.tpi30_p
USING gist (ST_ConvexHull(rast));

SELECT AddRasterConstraints('schema_marcinczyk'::name,
'tpi30'::name,'rast'::name);

SELECT st_summarystats(tpi.rast)
FROM schema_marcinczyk.tpi30_p AS tpi;
```

	st_summarystats 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)
8	(10000,0,0,1.1461852489890105,-8.25,8.625)

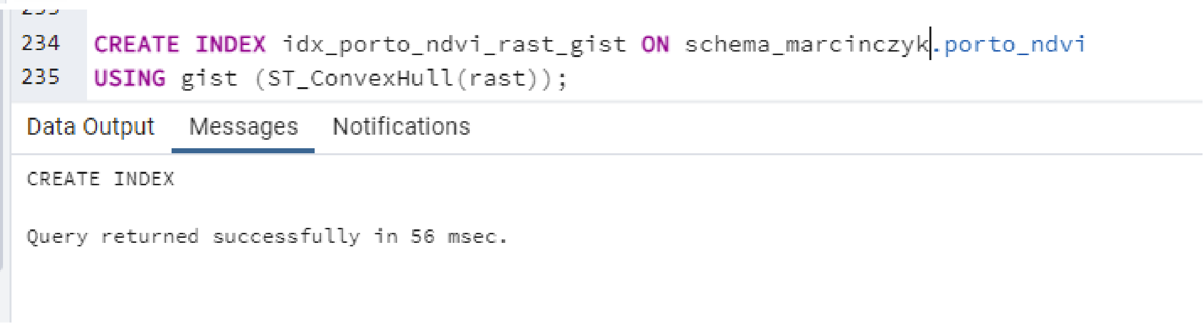
Algebra map

Przykład 1 - Wyrażenie Algebry Map



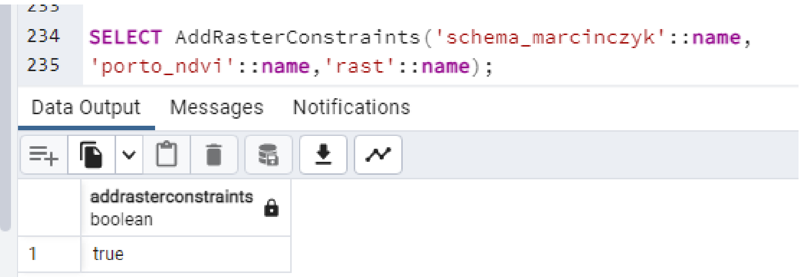
```sql
CREATE TABLE schema_marcinczyk.porto_ndvi AS
WITH r AS (
SELECT a.rid,ST_Clip(a.rast, b.geom,true) AS rast
FROM rasters.landsat8 AS a, vectors.porto_parishes AS b
WHERE b.municipality ilike 'porto' and ST_Intersects(b.geom,a.rast)
)
SELECT
r.rid,ST_MapAlgebra(
r.rast, 1,
r.rast, 4,
'([rast2.val] - [rast1.val]) / ([rast2.val] +
[rast1.val])::float','32BF'
) AS rast
FROM r;

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

	st_summarystats summarystats
1	(3441,472.5628735020291,0.13733300595816017,0.08585895884496325,-0.05336048826575279,0.5395634770393372)
2	(1594,405.59193430282176,0.2544491432263625,0.10302680565556493,0.014258482493460178,0.510151743888855)
3	(7342,1281.4318716040143,0.17453444178752578,0.10410533725920618,-0.05492142215371132,0.535823762416839...
4	(358,58.56076838821173,0.16357756533020035,0.05807487011744044,0.04480135068297386,0.3640957772731781)
5	(6114,922.638408873936,0.15090585686521688,0.08509527173750826,-0.06950689852237701,0.4920217990875244)
6	(6949,954.9930231836624,0.13742884201808353,0.07167064207589297,-0.007534984033554792,0.465824574232101...
7	(5,0.5146020501852036,0.10292041003704071,0.006468244648940868,0.09488994628190994,0.11467690765857697)
8	(7961,1173.9152122675441,0.14745826055364203,0.06630593157807599,-0.04501929506659508,0.518858790397644)
9	(6035,686.2183526008739,0.11370643787918375,0.07157472096085872,-0.055844053626060486,0.559989094734191...
10	(1150,160.76246094996168,0.1397934443043145,0.07185332833699404,-0.03956645727157593,0.4340870976448059)
11	(5050,649.9646308438387,0.12870586749382945,0.08776472180579241,-0.05340314283967018,0.5274097919464111)



```sql
CREATE INDEX idx_porto_ndvi_rast_gist ON schema_marcinczyk.porto_ndvi
USING gist (ST_ConvexHull(rast));
```

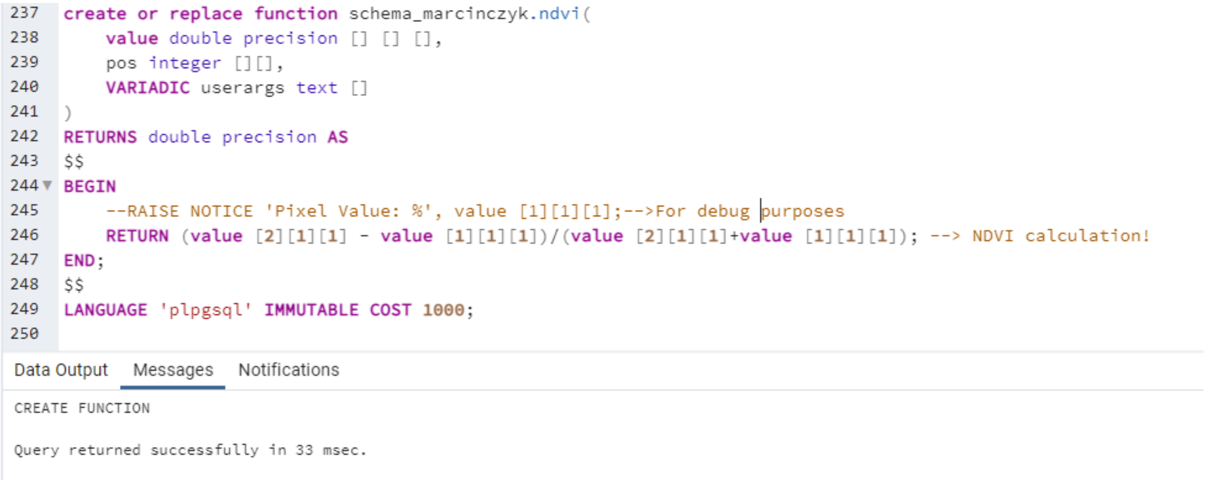
```
CREATE INDEX

Query returned successfully in 56 msec.
```



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

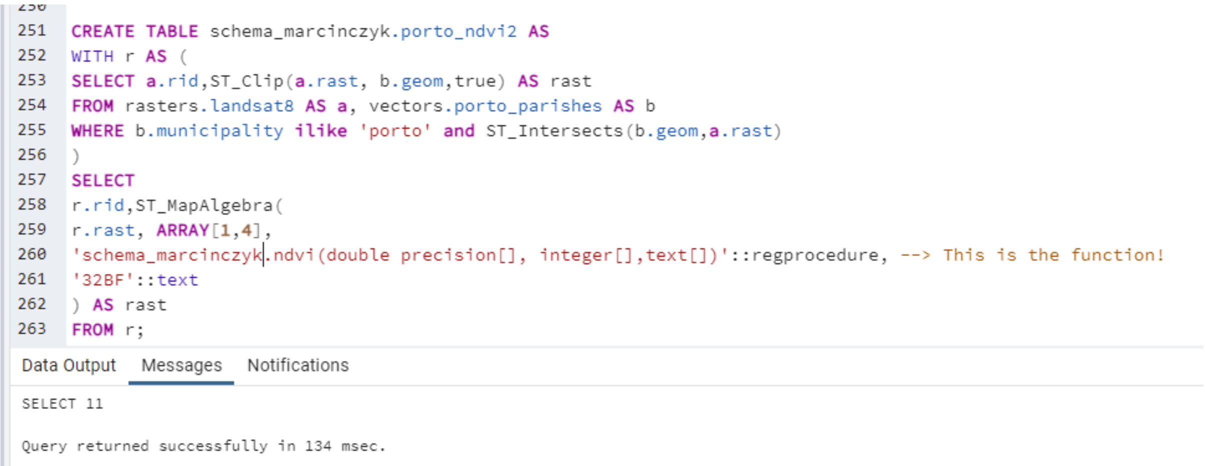
	addrasterconstraints boolean
1	true

Przykład 2 - Funkcja zwrotna



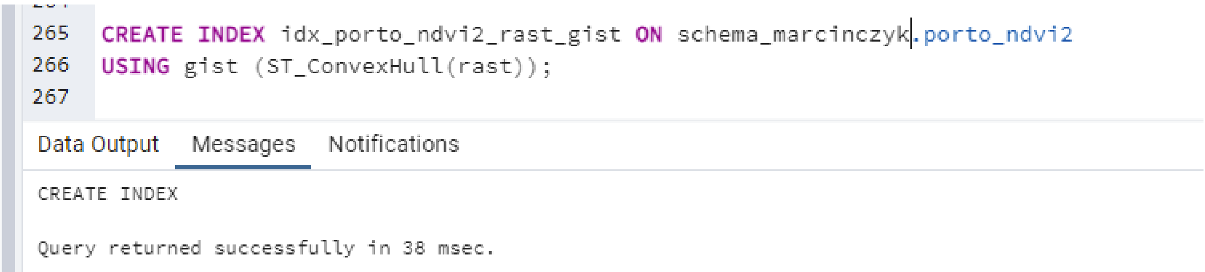
```
create or replace function schema_marcinczyk.ndvi(
    value double precision [] [] [],
    pos integer [][],
    VARIADIC userargs text []
)
RETURNS double precision AS
$$
BEGIN
    --RAISE NOTICE 'Pixel Value: %', value [1][1][1];-->For debug purposes
    RETURN (value [2][1][1] - value [1][1][1])/(value [2][1][1]+value [1][1][1]); --> NDVI calculation!
END;
$$
LANGUAGE 'plpgsql' IMMUTABLE COST 1000;
```

```
CREATE FUNCTION

Query returned successfully in 33 msec.
```



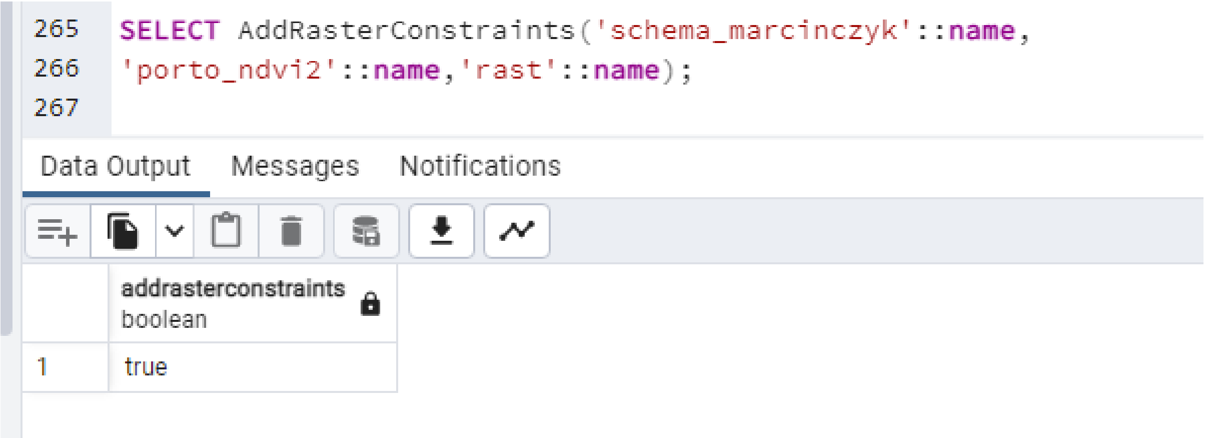
```
CREATE TABLE schema_marcinczyk.porto_ndvi2 AS
WITH r AS (
SELECT a.rid,ST_Clip(a.rast, b.geom,true) AS rast
FROM rasters.landsat8 AS a, vectors.porto_parishes AS b
WHERE b.municipality ilike 'porto' and ST_Intersects(b.geom,a.rast)
)
SELECT
r.rid,ST_MapAlgebra(
r.rast, ARRAY[1,4],
'schema_marcinczyk.ndvi(double precision[], integer[],text[])'::regprocedure, --> This is the function!
'32BF'::text
) AS rast
FROM r;
```

```
SELECT 11

Query returned successfully in 134 msec.
```



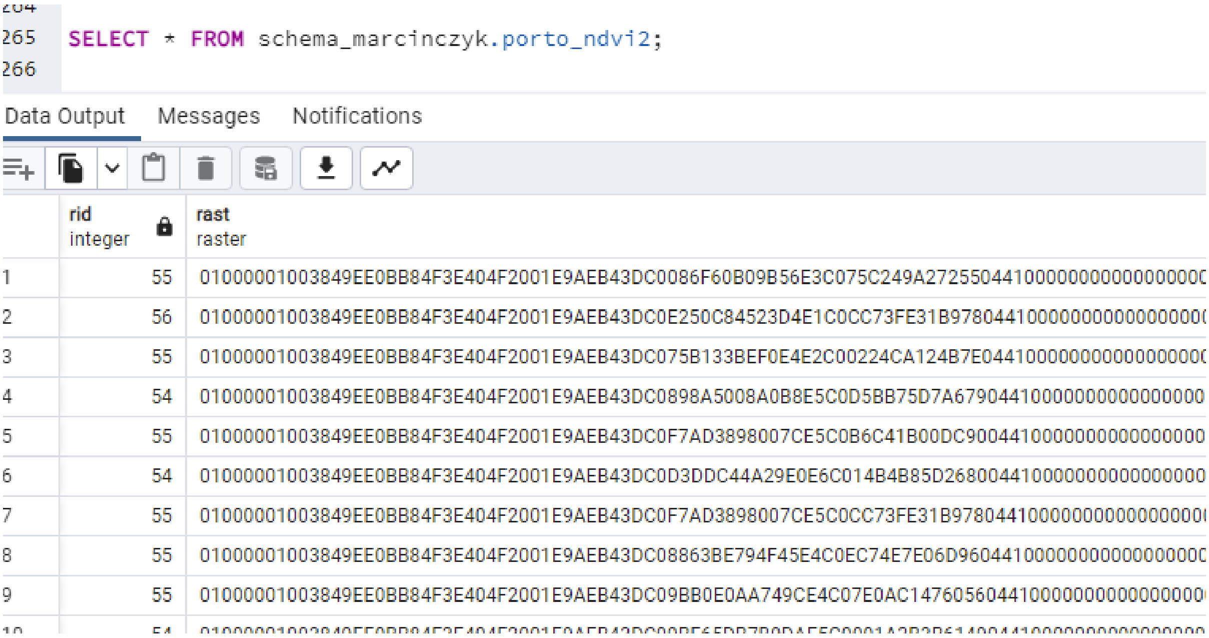
```
CREATE INDEX idx_porto_ndvi2_rast_gist ON schema_marcinczyk.porto_ndvi2
USING gist (ST_ConvexHull(rast));
```

```
CREATE INDEX

Query returned successfully in 38 msec.
```



```
SELECT AddRasterConstraints('schema_marcinczyk'::name,
'porto_ndvi2'::name,'rast'::name);
```

	addrasterconstraints boolean
1	true

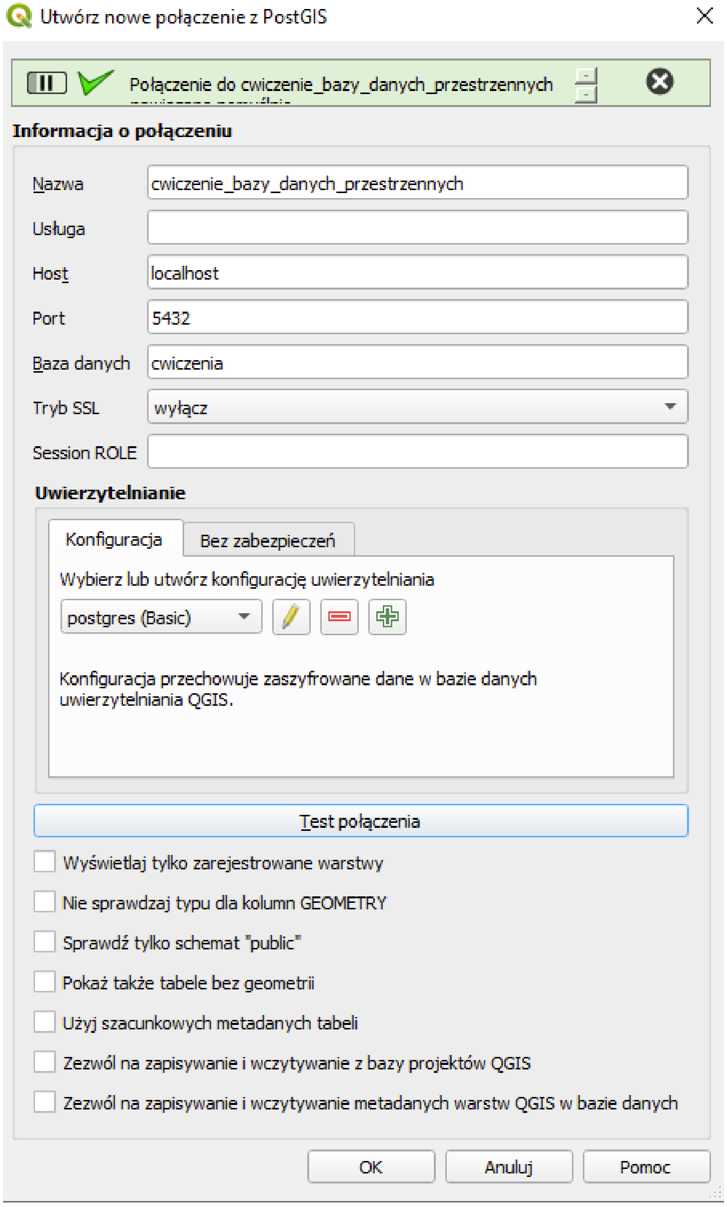


```sql
SELECT * FROM schema_marcinczyk.porto_ndvi2;
```

	rid integer	rast raster
1	55	01000001003849EE0BB84F3E404F2001E9AEB43DC0086F60B09B56E3C075C249A272550441000000000000000000
2	56	01000001003849EE0BB84F3E404F2001E9AEB43DC0E250C84523D4E1C0CC73FE31B978044100000000000000000
3	55	01000001003849EE0BB84F3E404F2001E9AEB43DC075B133BEF0E4E2C00224CA124B7E044100000000000000000
4	54	01000001003849EE0BB84F3E404F2001E9AEB43DC0898A5008A0B8E5C0D5BB75D7A67904410000000000000000
5	55	01000001003849EE0BB84F3E404F2001E9AEB43DC0F7AD3898007CE5C0B6C41B00DC9004410000000000000000
6	54	01000001003849EE0BB84F3E404F2001E9AEB43DC0D3DDC44A29E0E6C014B4B85D268004410000000000000000
7	55	01000001003849EE0BB84F3E404F2001E9AEB43DC0F7AD3898007CE5C0CC73FE31B978044100000000000000000
8	55	01000001003849EE0BB84F3E404F2001E9AEB43DC08863BE794F45E4C0EC74E7E06D9604410000000000000000
9	55	01000001003849EE0BB84F3E404F2001E9AEB43DC09BB0E0AA749CE4C07E0AC147605604410000000000000000
10	54	01000001003849EE0BB84F3E404F2001E9AEB43DC09BE65DB7B0DAE5C0001A2B3B6140044100000000000000000

Eksport danych

Przykład 0 - QGIS



Utwórz nowe połączenie z PostGIS

Połączenie do cwiczenie_bazy_danych_przestrzennych

Informacja o połączeniu

Nazwa	cwiczenie_bazy_danych_przestrzennych
Usługa	
Host	localhost
Port	5432
Baza danych	cwiczenia
Tryb SSL	wyłącz
Session ROLE	

Uwierzytelnianie

Konfiguracja
Bez zabezpieczeń

Wybierz lub utwórz konfigurację uwierzytelniania

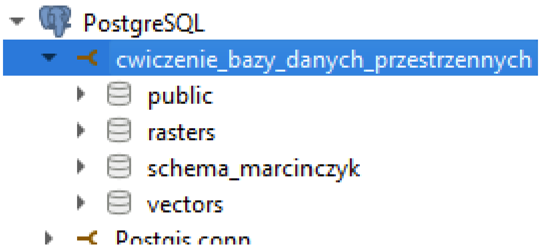
postgres (Basic)

Konfiguracja przechowuje zaszyfrowane dane w bazie danych uwierzytelniania QGIS.

Test połączenia

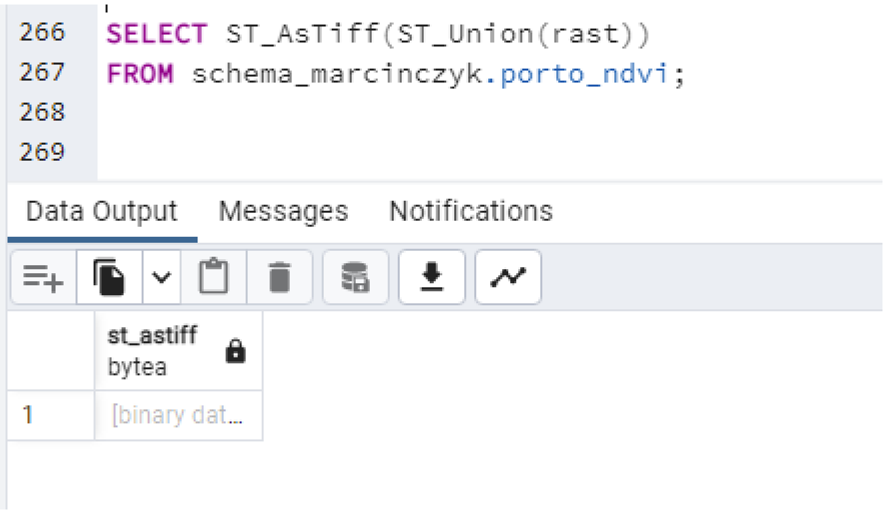
- ☐ Wyświetlaj tylko zarejestrowane warstwy
- ☐ Nie sprawdzaj typu dla kolumn GEOMETRY
- ☐ Sprawdź tylko schemat "public"
- ☐ Pokaż także tabele bez geometrii
- ☐ Użyj szacunkowych metadanych tabeli
- ☐ Zezwól na zapisywanie i wczytywanie z bazy projektów QGIS
- ☐ Zezwól na zapisywanie i wczytywanie metadanych warstw QGIS w bazie danych

OK
Anuluj
Pomoc



```
PostgreSQL
  cwiczenie_bazy_danych_przestrzennych
    public
    rasters
    schema_marcinczyk
    vectors
  Postgis conn
```

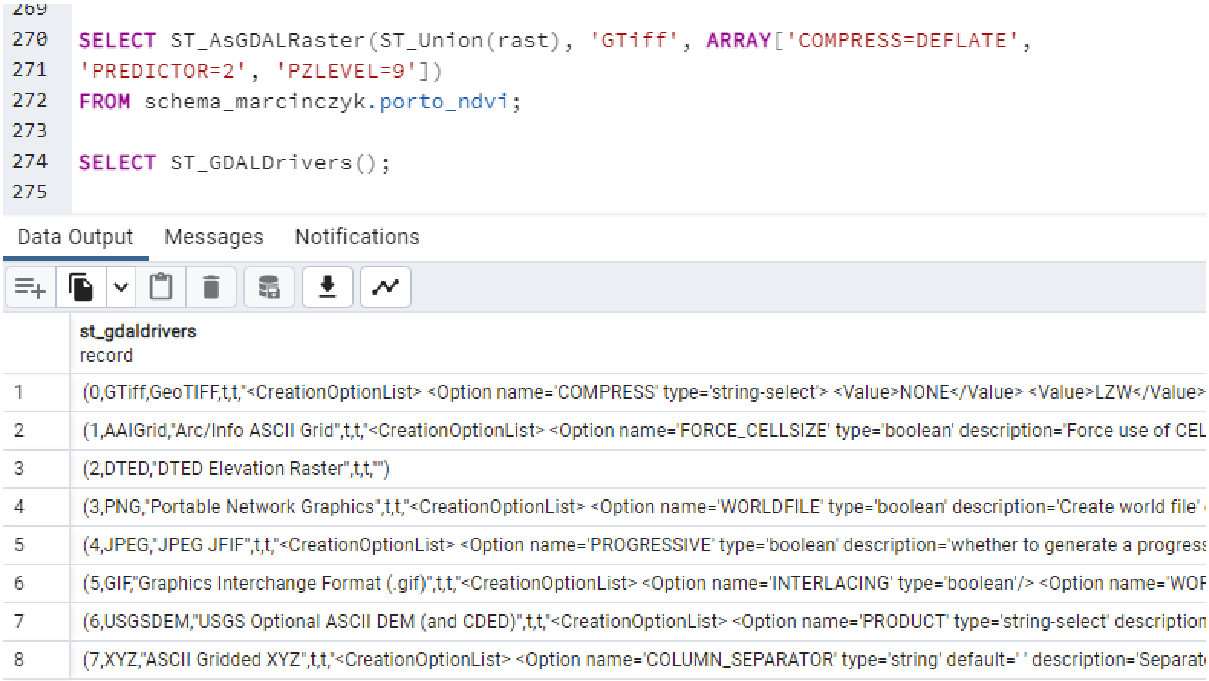
Przykład 1 - ST_AsTiff



```sql
SELECT ST_AsTiff(ST_Union(rast))
FROM schema_marcinczyk.porto_ndvi;
```

	st_astiff bytea
1	[binary dat...

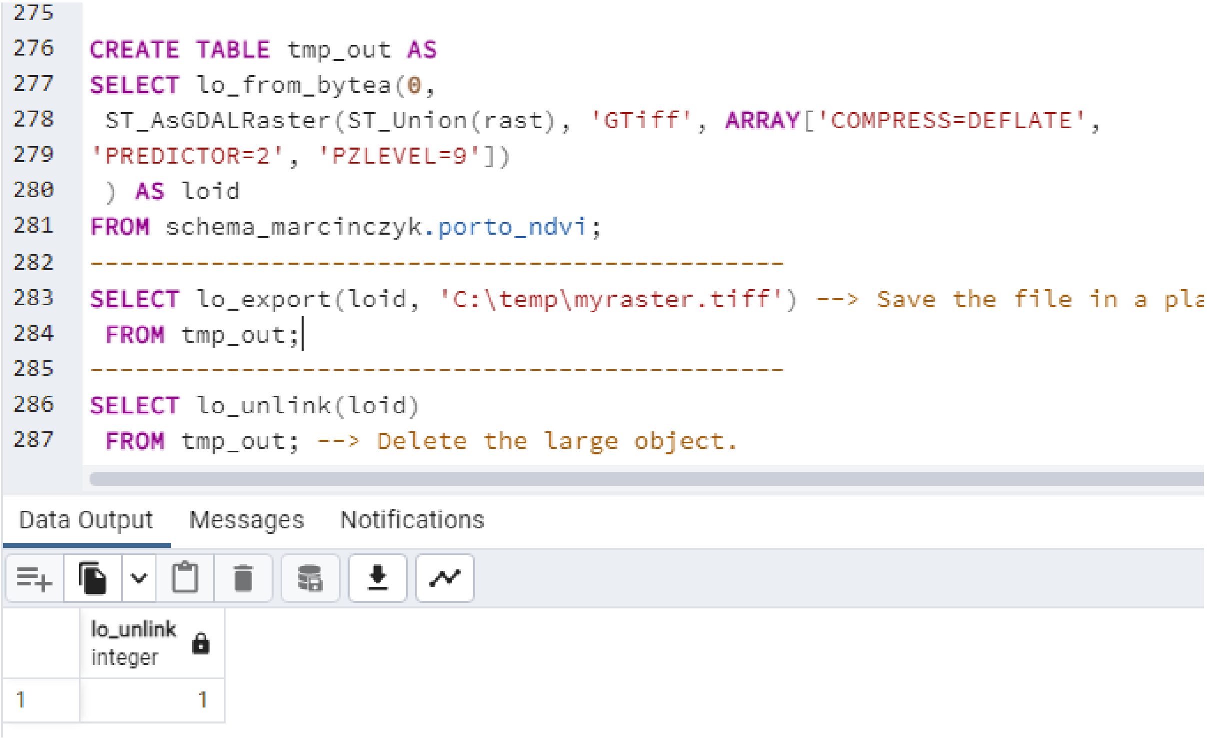
Przykład 2 - ST_AsGDALRaster



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

SELECT ST_GDALDrivers();
```

	st_gdaldrivers record
1	(0,GTiff,GeoTIFF,t,t,"<CreationOptionList> <Option name='COMPRESS' type='string-select'> <Value>NONE</Value> <Value>LZW</Value>
2	(1,AAIGrid,"Arc/Info ASCII Grid",t,t,"<CreationOptionList> <Option name='FORCE_CELLSIZE' type='boolean' description='Force use of CEL
3	(2,DTED,"DTED Elevation Raster",t,t,"")
4	(3,PNG,"Portable Network Graphics",t,t,"<CreationOptionList> <Option name='WORLDFILE' type='boolean' description='Create world file'
5	(4,JPEG,"JPEG JFIF",t,t,"<CreationOptionList> <Option name='PROGRESSIVE' type='boolean' description='whether to generate a progress
6	(5,GIF,"Graphics Interchange Format (.gif)",t,t,"<CreationOptionList> <Option name='INTERLACING' type='boolean'/> <Option name='WOF
7	(6,USGSDEM,"USGS Optional ASCII DEM (and CDED)",t,t,"<CreationOptionList> <Option name='PRODUCT' type='string-select' description
8	(7,XYZ,"ASCII Gridded XYZ",t,t,"<CreationOptionList> <Option name='COLUMN_SEPARATOR' type='string' default=' ' description='Separat

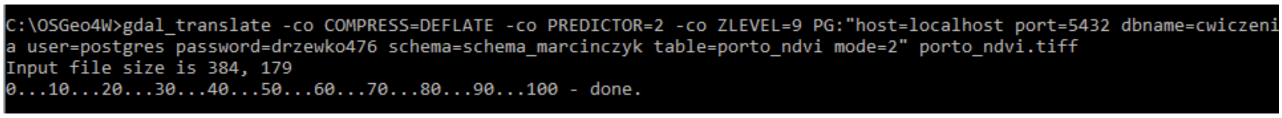
Przykład 3 - Zapisywanie danych na dysku za pomocą dużego obiektu (large object, lo)



```

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 schema_marcinczyk.porto_ndvi;
----------------------------------------------
SELECT lo_export(loid, 'C:\temp\myraster.tiff') --> Save the file in a pla
 FROM tmp_out;
----------------------------------------------
SELECT lo_unlink(loid)
 FROM tmp_out; --> Delete the large object.
```

	lo_unlink integer
1	1

Przykład 4 - Użycie GDAL



```
C:\OSGeo4W>gdal_translate -co COMPRESS=DEFLATE -co PREDICTOR=2 -co ZLEVEL=9 PG:"host=localhost port=5432 dbname=cwiczeni
a user=postgres password=drzewko476 schema=schema_marcinczyk 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.
```