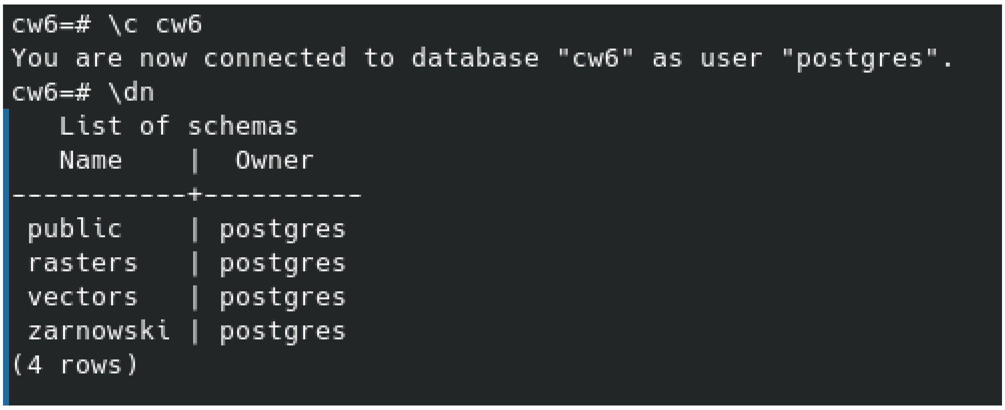
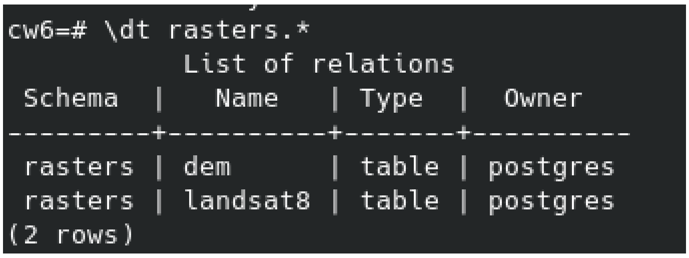
Konfiguracja

```
create database cw6;
create extension postgis;

pg_restore -d cw6 -l
/home/krzysztof/Downloads/postgis_raster.backup

jdbc:postgresql://localhost:5432/cw6

alter schema schema_name rename to zarnowski;
```



```
cw6=# \c cw6
You are now connected to database "cw6" as user "postgres".
cw6=# \dn
   List of schemas
   Name    |  Owner
-----------+----------
 public    | postgres
 rasters   | postgres
 vectors   | postgres
 zarnowski | postgres
(4 rows)
```

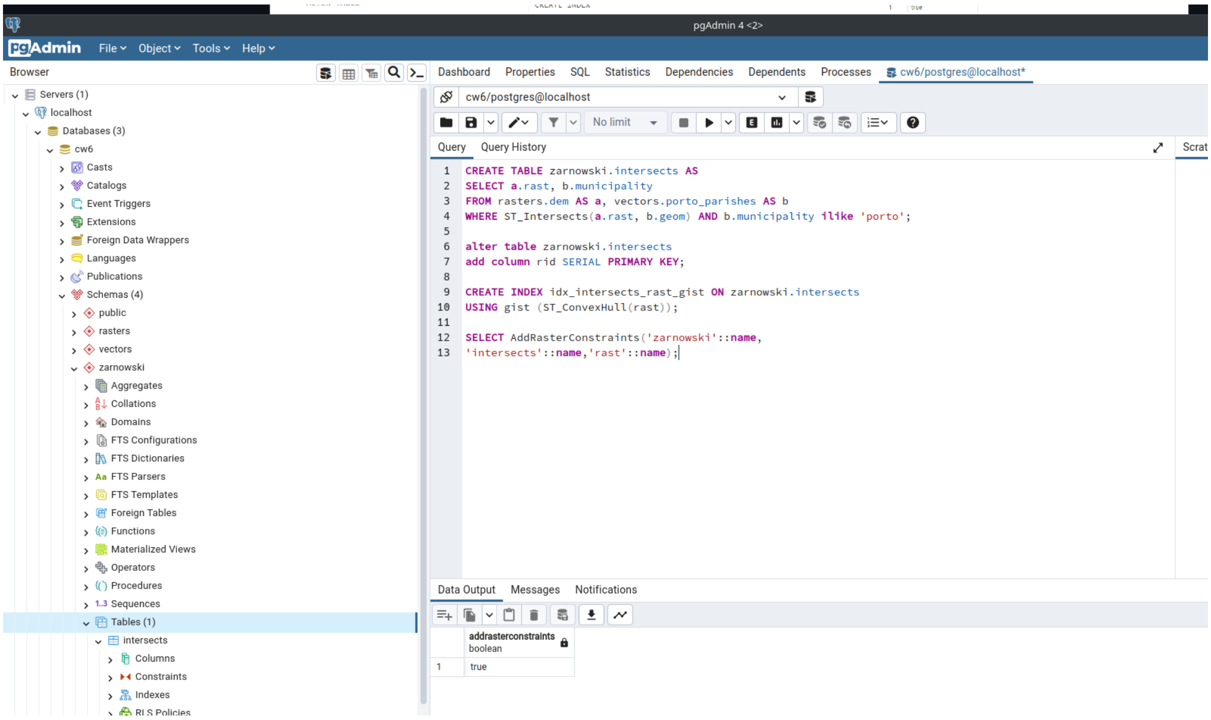
Załadowanie danych rastrowych



```
cw6=# \dt rasters.*
           List of relations
 Schema  |   Name   | Type  |  Owner
---------+----------+-------+----------
 rasters | dem      | table | postgres
 rasters | landsat8 | table | postgres
(2 rows)
```

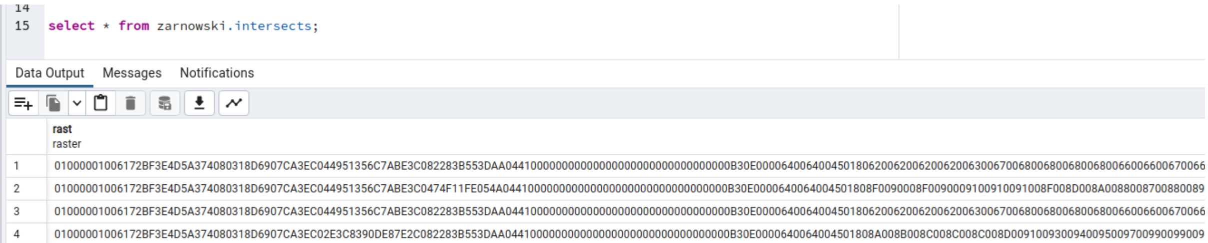
Tworzenie rastrów z istniejących rastrów i interakcja z wektorami

Przykład 1 - ST_Intersects



```
CREATE TABLE zarnowski.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';

alter table zarnowski.intersects
add column rid SERIAL PRIMARY KEY;

CREATE INDEX idx_intersects_rast_gist ON zarnowski.intersects
USING gist (ST_ConvexHull(rast));

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

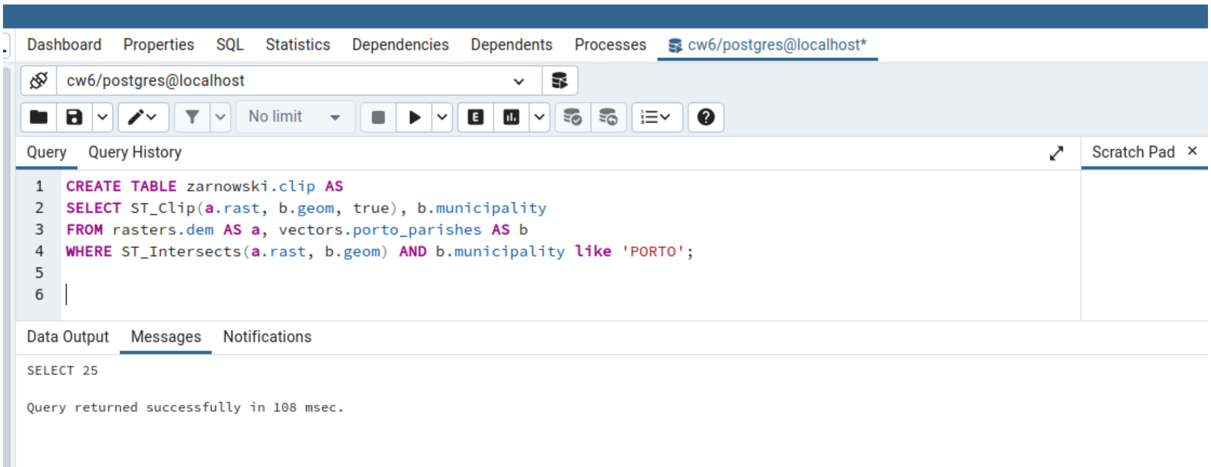
	addrasterconstraints boolean
1	true



```
select * from zarnowski.intersects;
```

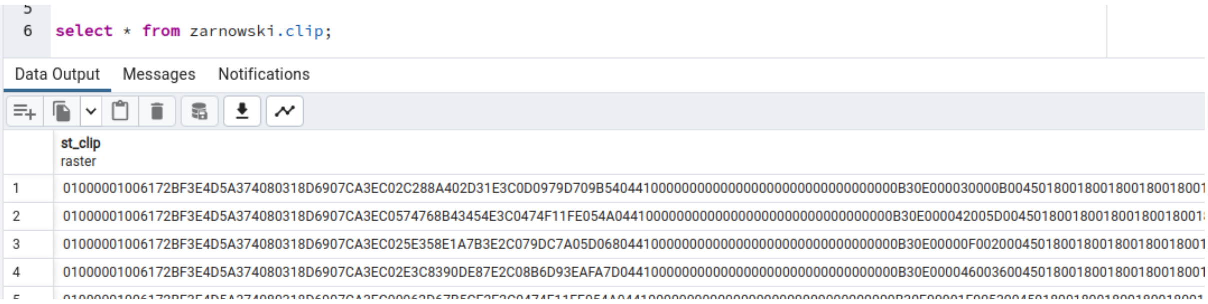
	rast raster
1	01000001006172BF3E4D5A374080318D6907CA3EC044951356C7ABE3C082283B553DAA044100000000000000000000000000000000B30E00006400640045018062006200620062006300670068006800680068006600660067006
2	01000001006172BF3E4D5A374080318D6907CA3EC044951356C7ABE3C0474F11FE054A044100000000000000000000000000000000B30E000064006400450180 8F0090008F009000910091 0091008F008D008A0088008700880089
3	01000001006172BF3E4D5A374080318D6907CA3EC044951356C7ABE3C082283B553DAA044100000000000000000000000000000000B30E00006400640045018062006200620062006300670068006800680068006600660067006
4	01000001006172BF3E4D5A374080318D6907CA3EC02E3C8390DE87E2C082283B553DAA044100000000000000000000000000000000B30E0000640064004501808A008B008C008C008C008D009100930094009500970099009

Przykład 2 - ST_Clip


Dashboard Properties SQL Statistics Dependencies Dependents Processes cw6/postgres@localhost*
cw6/postgres@localhost
No limit
Query Query History
Scratch Pad ×
Data Output Messages Notifications

```
CREATE TABLE zarnowski.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';

```

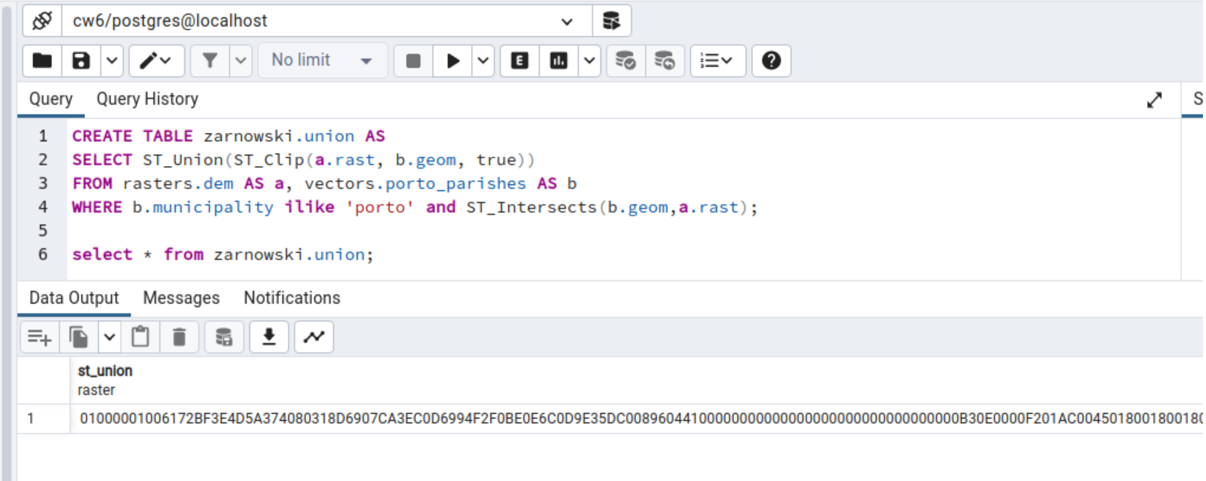
```
SELECT 25

Query returned successfully in 108 msec.
```



```
select * from zarnowski.clip;
```

```
  st_clip
  raster
1 01000001006172BF3E4D5A374080318D6907CA3EC02C288A402D31E3C0D0979D709B54044100000000000000000000000000000000B30E000030000B00450180018001800180018001
2 01000001006172BF3E4D5A374080318D6907CA3EC0574768B43454E3C0474F11FE054A044100000000000000000000000000000000B30E000042005D00450180018001800180018001
3 01000001006172BF3E4D5A374080318D6907CA3EC025E358E1A7B3E2C079DC7A05D068044100000000000000000000000000000000B30E00000F00200045018001800180018001800
4 01000001006172BF3E4D5A374080318D6907CA3EC02E3C8390DE87E2C08B6D93EAFA7D044100000000000000000000000000000000B30E0000460036004501800180018001800180
5 01000001006172BF3E4D5A374080318D6907CA3EC00062D67BECE2E2C0474F11FE054A044100000000000000000000000000000000B30E00001F005300450180018001800180018001
```

Przykład 3 - St_Union

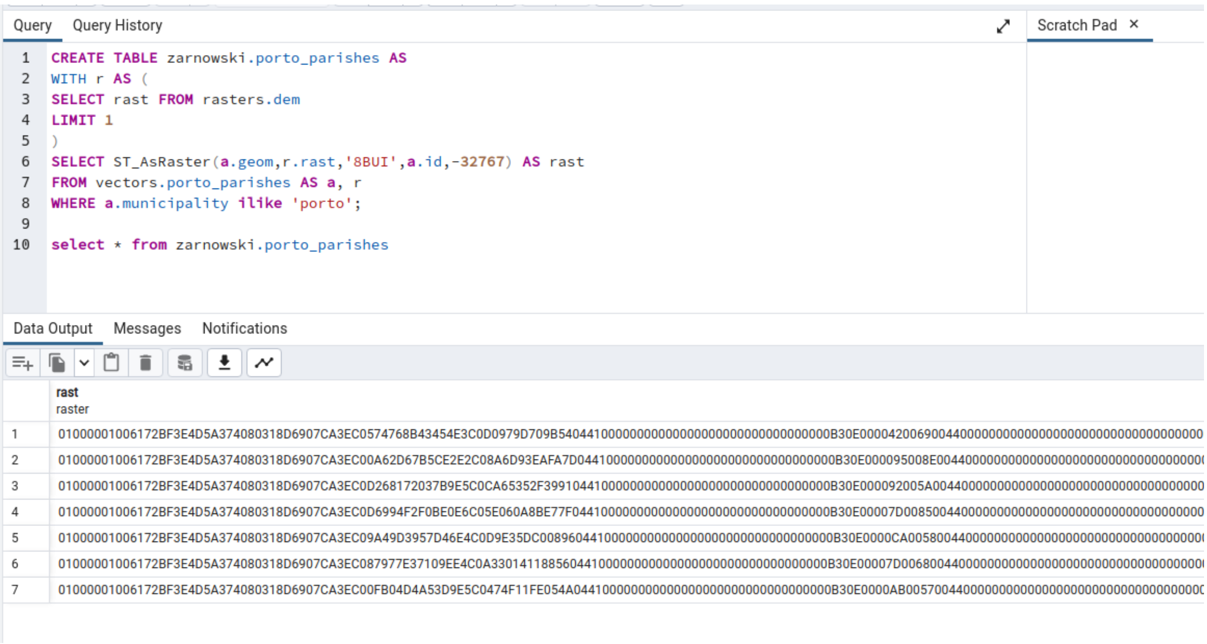


```
CREATE TABLE zarnowski.union AS
SELECT ST_Union(ST_Clip(a.rast, b.geom, true))
FROM rasters.dem AS a, vectors.porto_parishes AS b
WHERE b.municipality ilike 'porto' and ST_Intersects(b.geom,a.rast);

select * from zarnowski.union;
```

```
  st_union
  raster
1 01000001006172BF3E4D5A374080318D6907CA3EC0D6994F2F0BE0E6C0D9E35DC008960441000000000000000000000000000000000B30E0000F201AC004501800180018
```

Tworzenie rastrów z wektorów (rastrowanie)

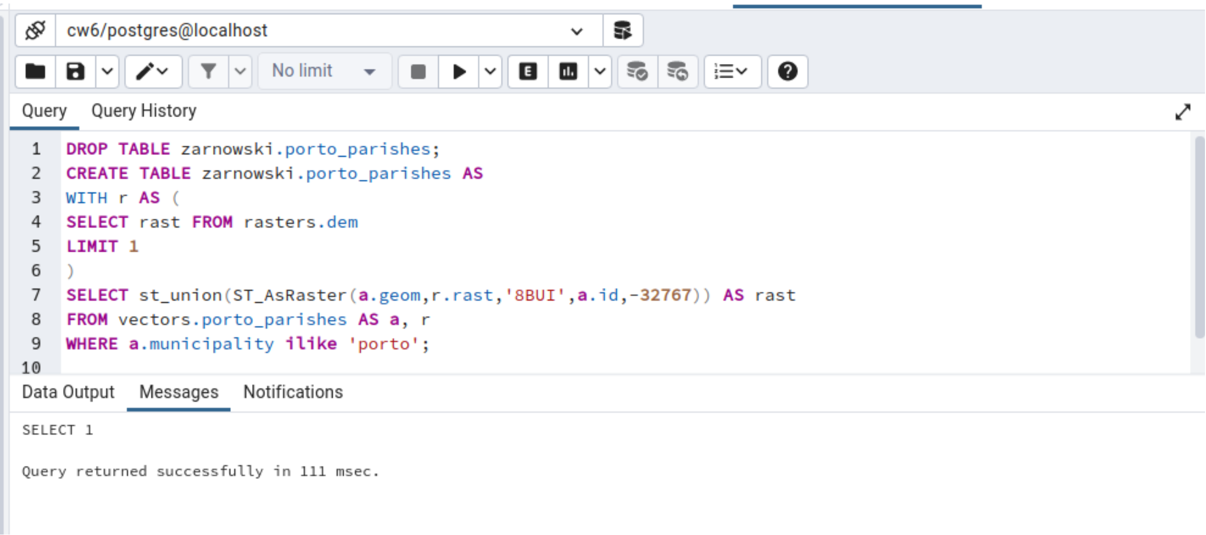
Przykład 1 - ST_AsRaster



```
CREATE TABLE zarnowski.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 * from zarnowski.porto_parishes
```

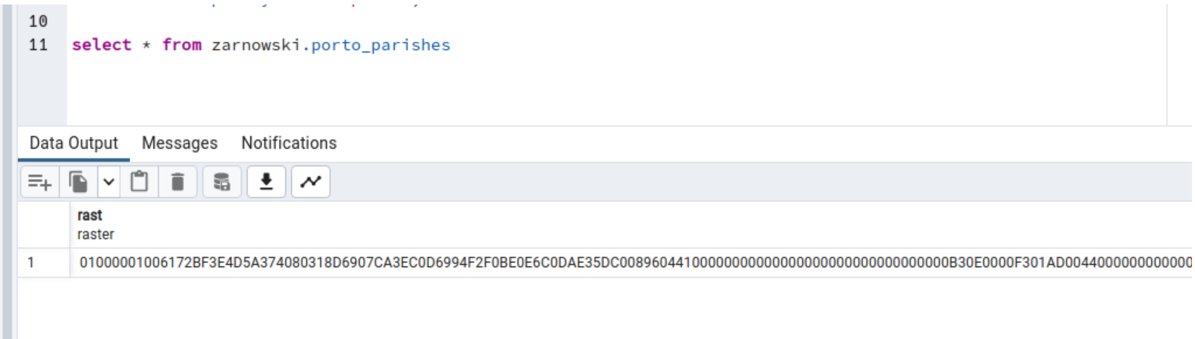
```
  rast
  raster
1 01000001006172BF3E4D5A374080318D6907CA3EC0574768B43454E3C0D0979D709B54044100000000000000000000000000000000B30E0000420069004400000000000000000000000000000000000
2 01000001006172BF3E4D5A374080318D6907CA3EC00A62D67B5CE2E2C08A6D93EAFA7D044100000000000000000000000000000000B30E000095008E004400000000000000000000000000000000000
3 01000001006172BF3E4D5A374080318D6907CA3EC0D268172037B9E5C0CA65352F3991044100000000000000000000000000000000B30E000092005A004400000000000000000000000000000000000
4 01000001006172BF3E4D5A374080318D6907CA3EC0D6994F2F0BE0E6C05E060A8BE77F044100000000000000000000000000000000B30E00007D0085004400000000000000000000000000000000000
5 01000001006172BF3E4D5A374080318D6907CA3EC09A49D3957D46E4C0D9E35DC00896044100000000000000000000000000000000B30E0000CA0058004400000000000000000000000000000000000
6 01000001006172BF3E4D5A374080318D6907CA3EC087977E37109EE4C0A33014118856044100000000000000000000000000000000B30E00007D0068004400000000000000000000000000000000000
7 01000001006172BF3E4D5A374080318D6907CA3EC00FB04D4A53D9E5C0474F11FE054A044100000000000000000000000000000000B30E0000AB0057004400000000000000000000000000000000000
```

Przykład 2 - ST_Union



```
DROP TABLE zarnowski.porto_parishes;
CREATE TABLE zarnowski.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';
```

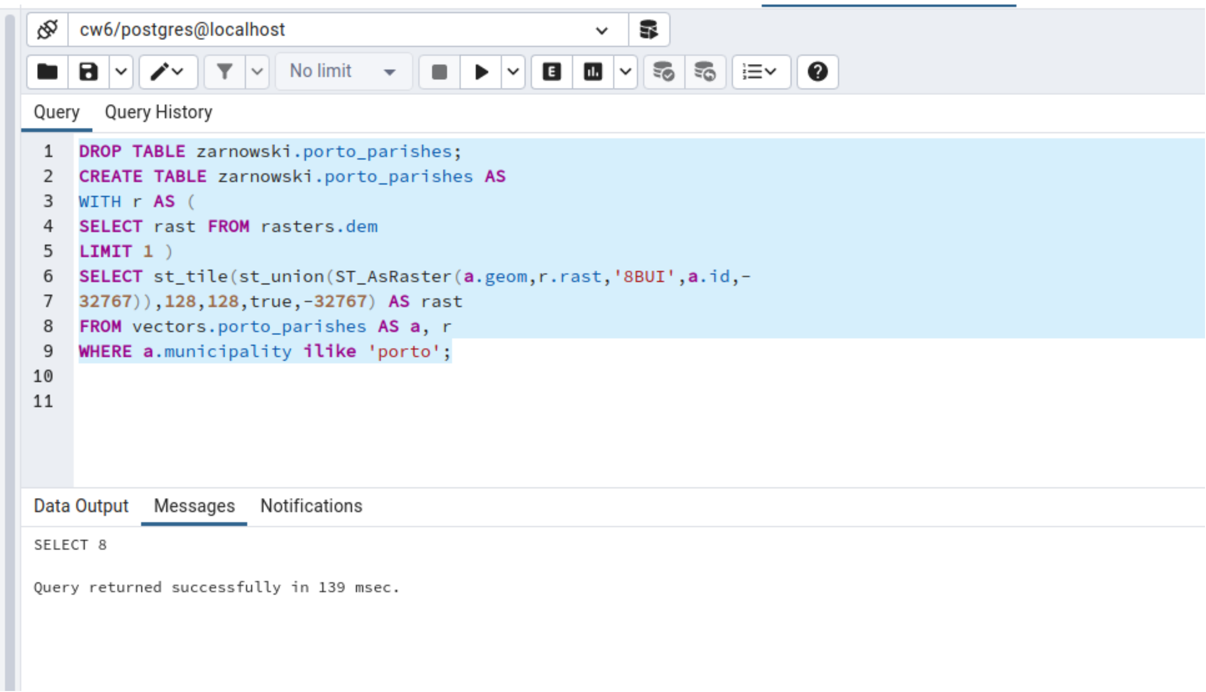
```
SELECT 1

Query returned successfully in 111 msec.
```



```
select * from zarnowski.porto_parishes
```

	rast raster
1	01000001006172BF3E4D5A374080318D6907CA3EC0D6994F2F0BE0E6C0DAE35DC00896044100000000000000000000000000000000B30E0000F301AD0044000000000000

Przykład 3 - ST_Tile

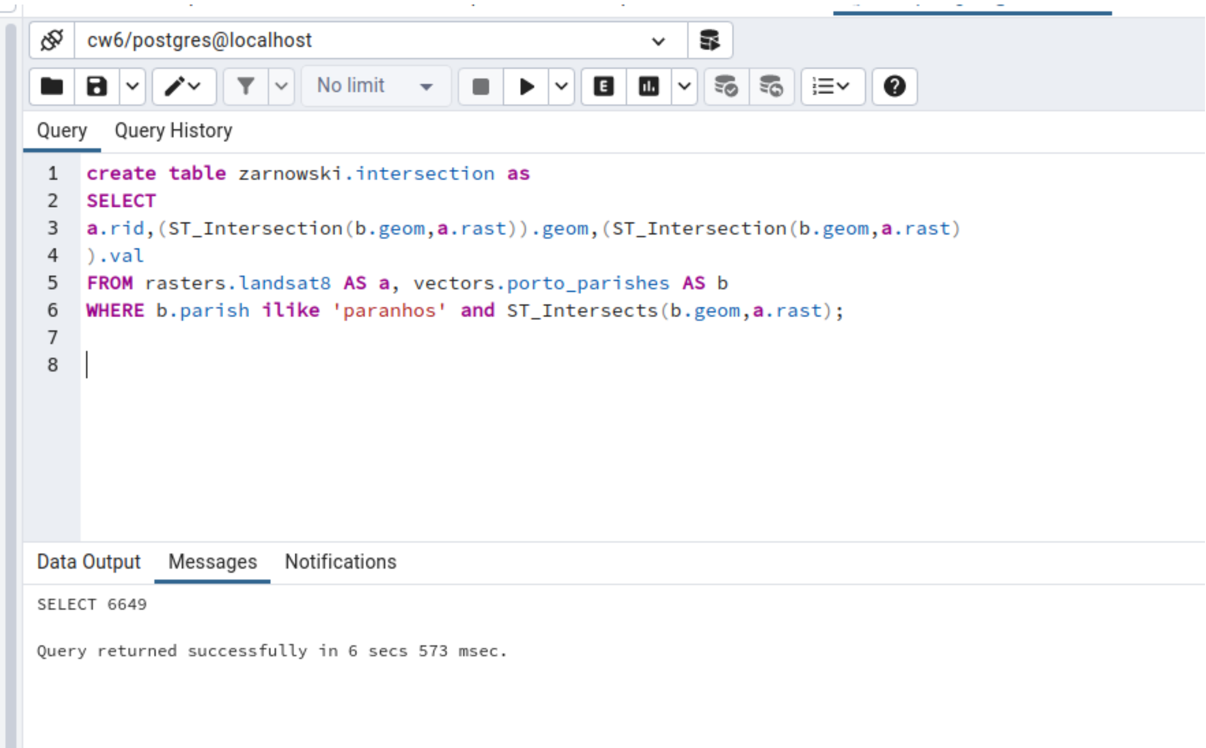


```
DROP TABLE zarnowski.porto_parishes;
CREATE TABLE zarnowski.porto_parishes AS
WITH r AS (
SELECT rast FROM rasters.dem
LIMIT 1 )
SELECT st_tile(st_union(ST_AsRaster(a.geom,r.rast,'8BUI',a.id,-
32767)),128,128,true,-32767) AS rast
FROM vectors.porto_parishes AS a, r
WHERE a.municipality ilike 'porto';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

```
SELECT 8

Query returned successfully in 139 msec.
```

Konwertowanie rastrów na wektory (wektoryzowanie)

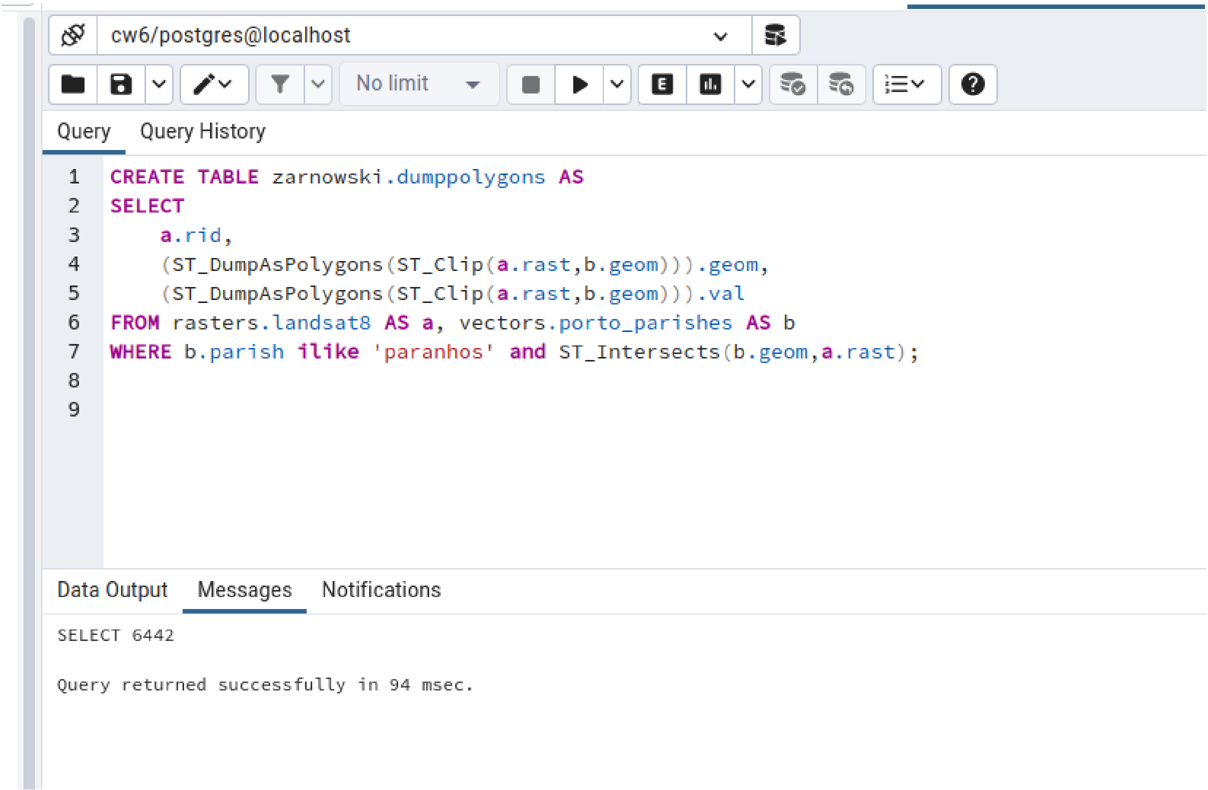
Przykład 1 – `ST_Intersection`



```
create table zarnowski.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 6649

Query returned successfully in 6 secs 573 msec.
```

Przykład 2 - `ST_DumpAsPolygons`

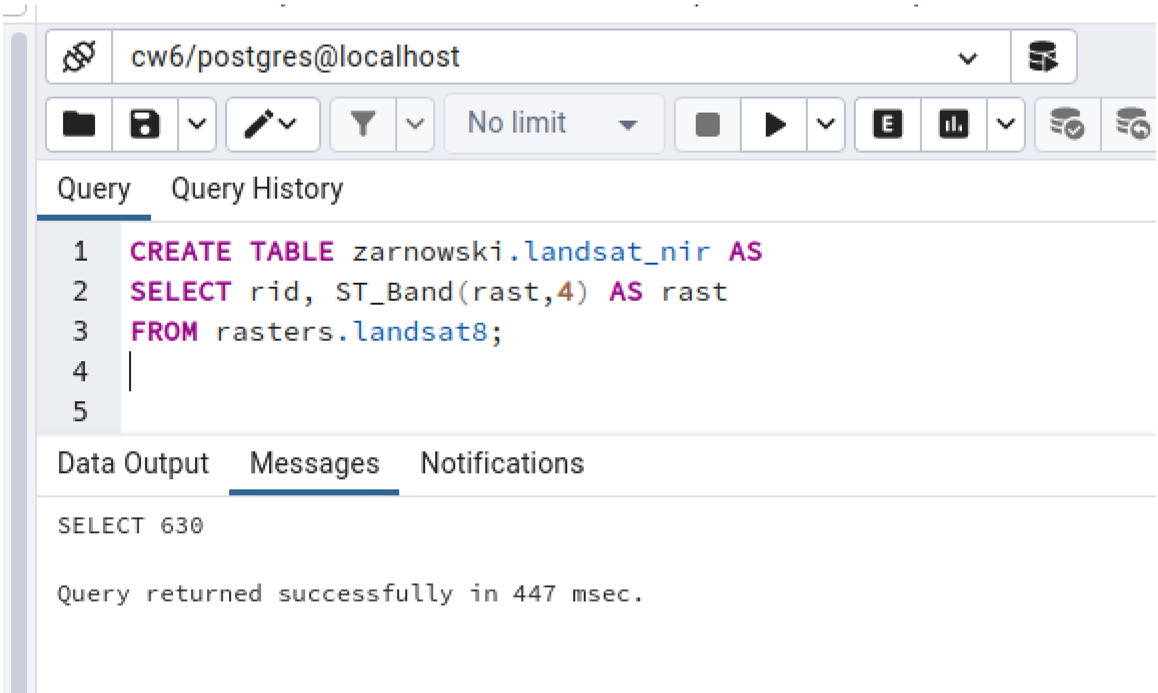


```
CREATE TABLE zarnowski.dumppolygons AS
SELECT
    a.rid,
    (ST_DumpAsPolygons(ST_Clip(a.rast,b.geom))).geom,
    (ST_DumpAsPolygons(ST_Clip(a.rast,b.geom))).val
FROM rasters.landsat8 AS a, vectors.porto_parishes AS b
WHERE b.parish ilike 'paranhos' and ST_Intersects(b.geom,a.rast);


```

```
SELECT 6442

Query returned successfully in 94 msec.
```

Analiza rastrów

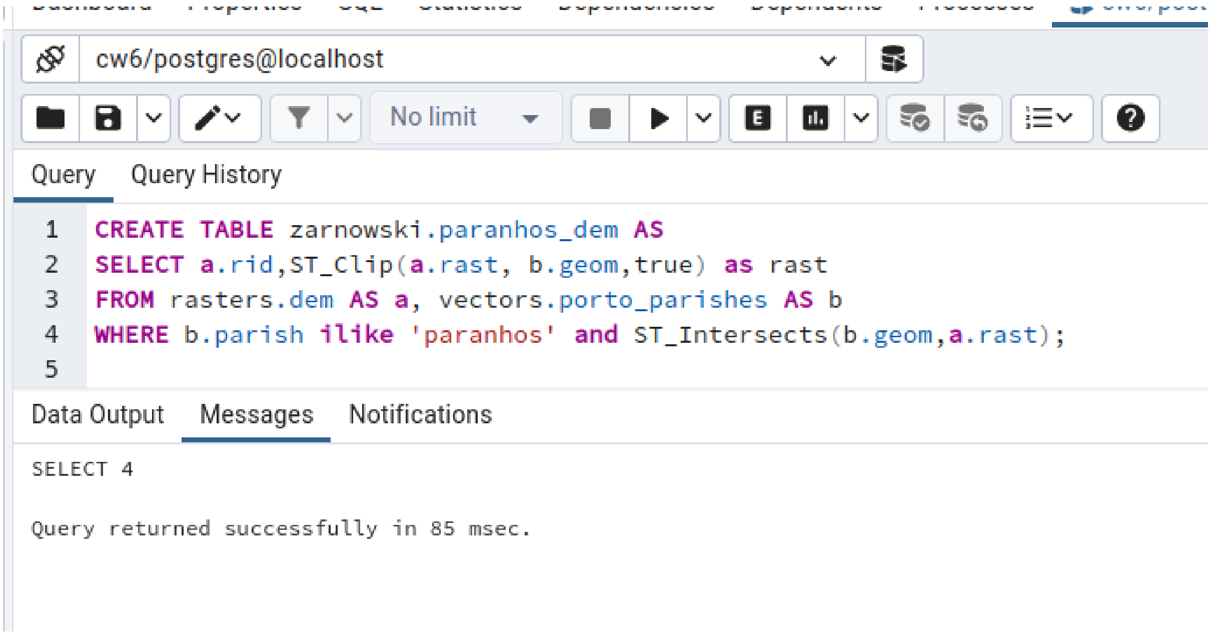
Przykład 1 - ST_Band



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

```
SELECT 630

Query returned successfully in 447 msec.
```

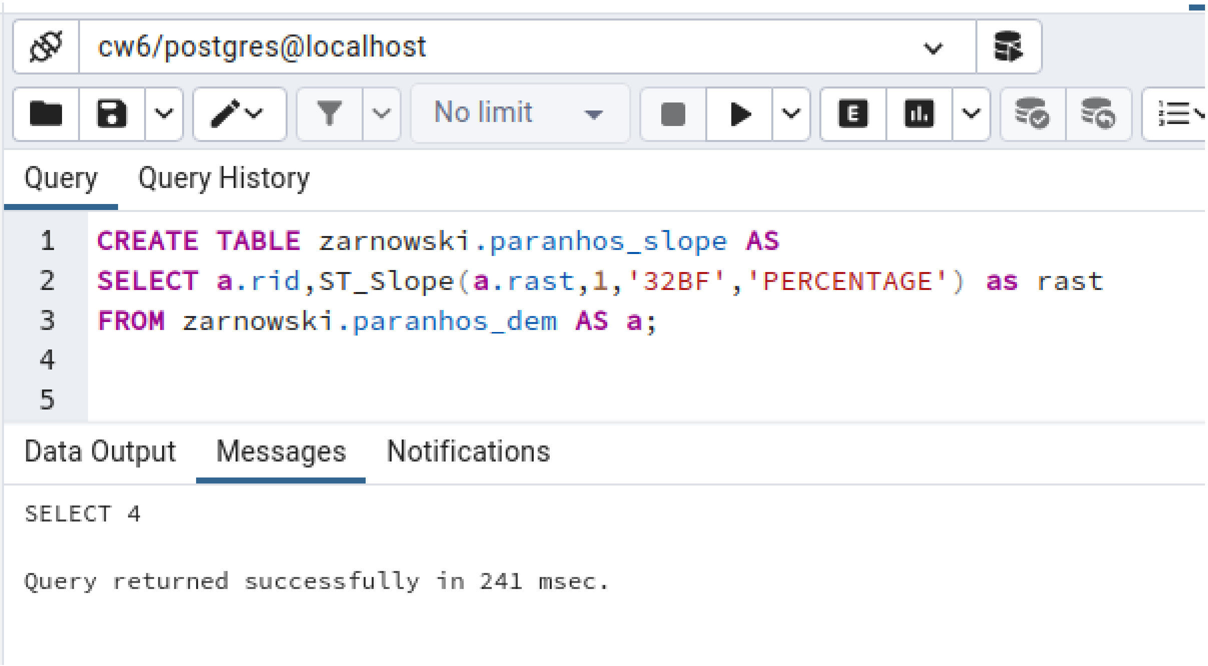
Przykład 2 - ST_Clip



```sql
CREATE TABLE zarnowski.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 4

Query returned successfully in 85 msec.
```

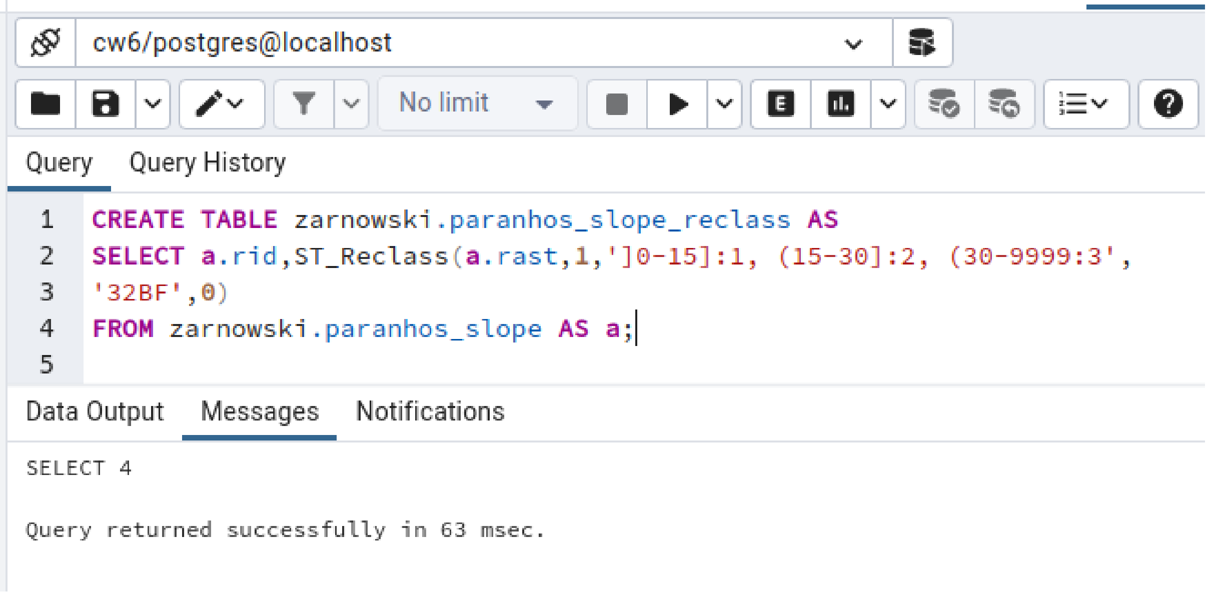
Przykład 3 - ST_Slope



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

```
SELECT 4

Query returned successfully in 241 msec.
```

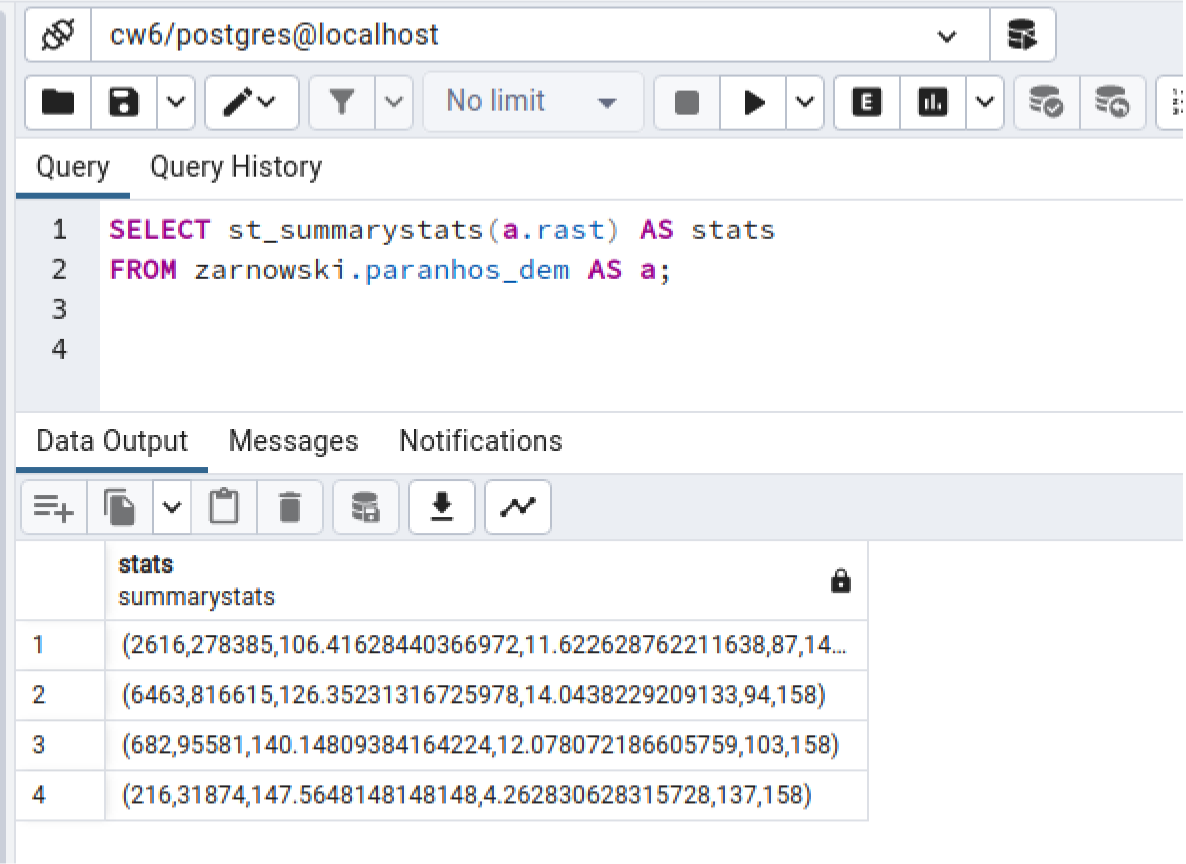
Przykład 4 - ST_Reclass



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

```
SELECT 4

Query returned successfully in 63 msec.
```

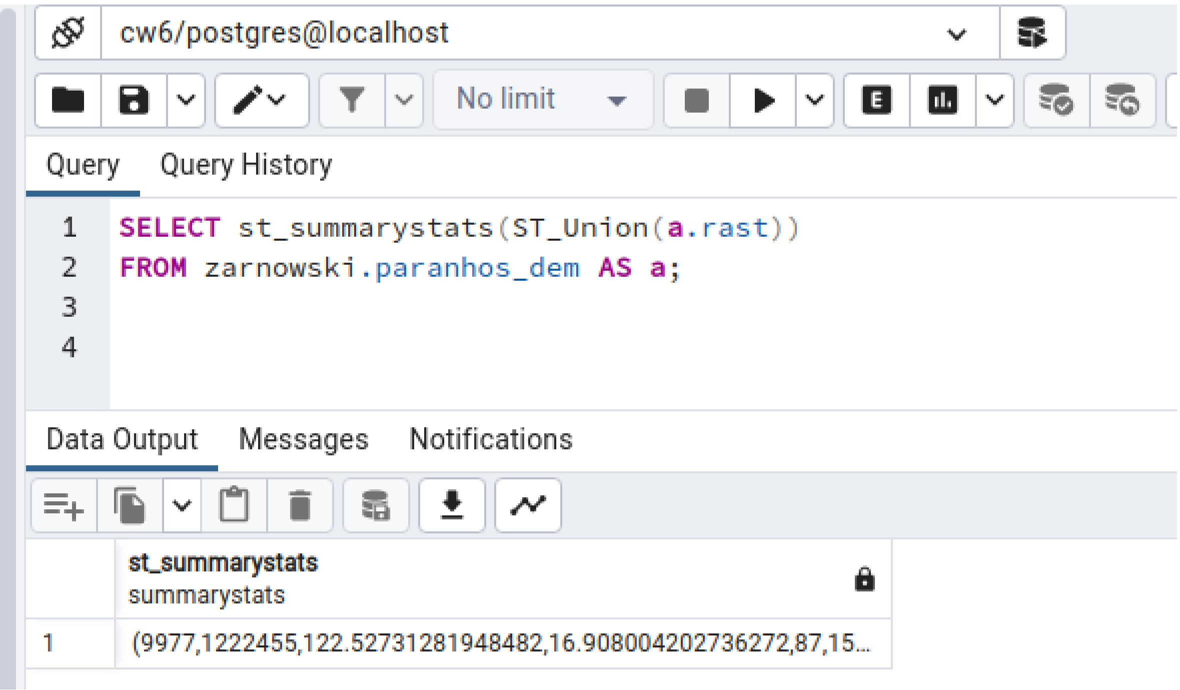
Przykład 5 - ST_SummaryStats



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

	stats summarystats
1	(2616,278385,106.41628440366972,11.622628762211638,87,14...
2	(6463,816615,126.35231316725978,14.0438229209133,94,158)
3	(682,95581,140.14809384164224,12.078072186605759,103,158)
4	(216,31874,147.5648148148148,4.262830628315728,137,158)

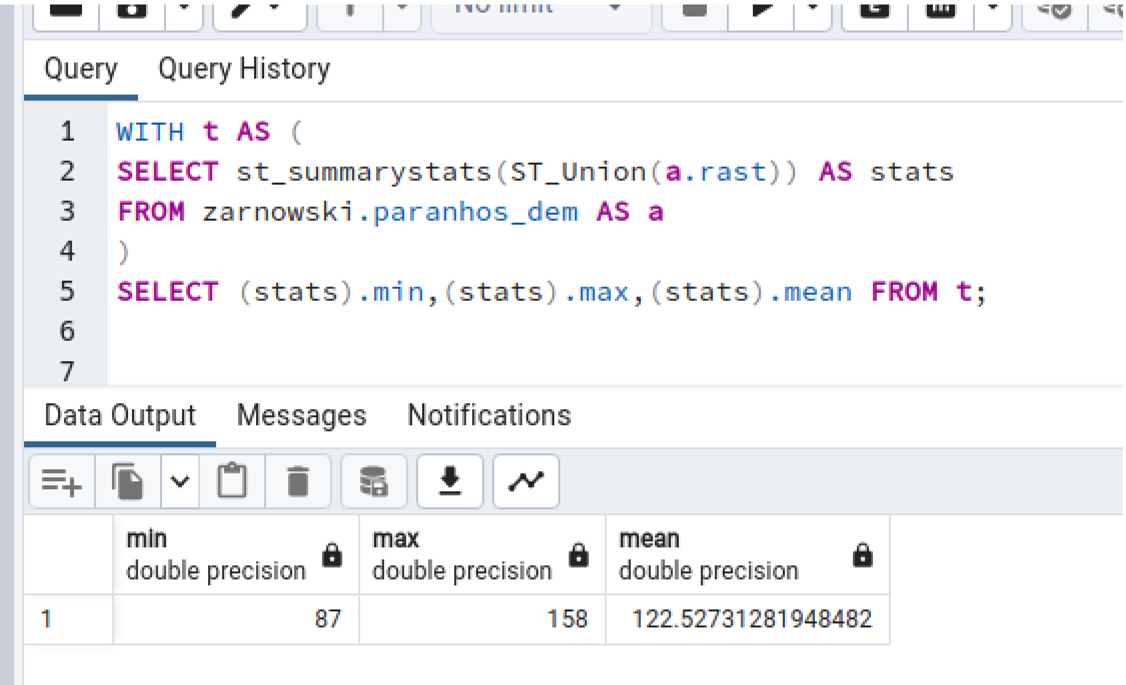
Przykład 6 - ST_SummaryStats oraz Union



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

	st_summarystats summarystats
1	(9977,1222455,122.52731281948482,16.908004202736272,87,15...

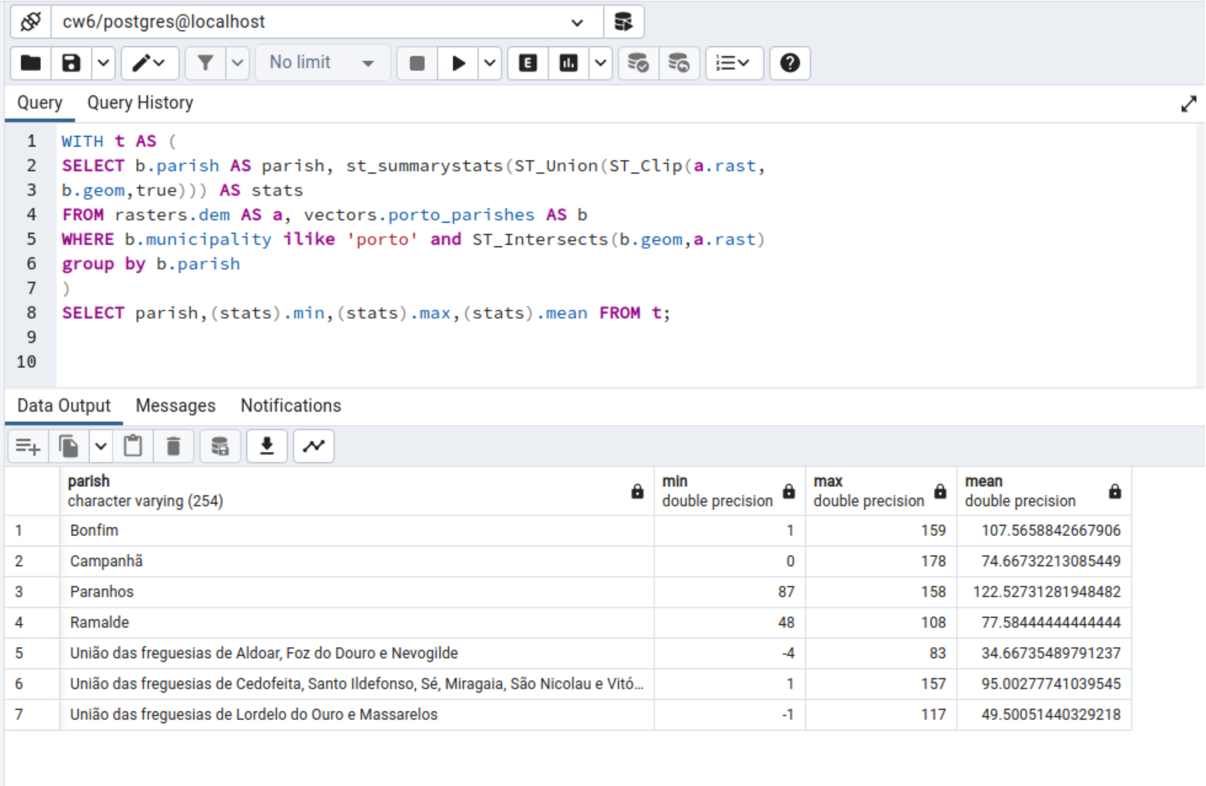
Przykład 7 - ST_SummaryStats z lepszą kontrolą złożonego typu danych



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

	min double precision	max double precision	mean double precision
1	87	158	122.52731281948482

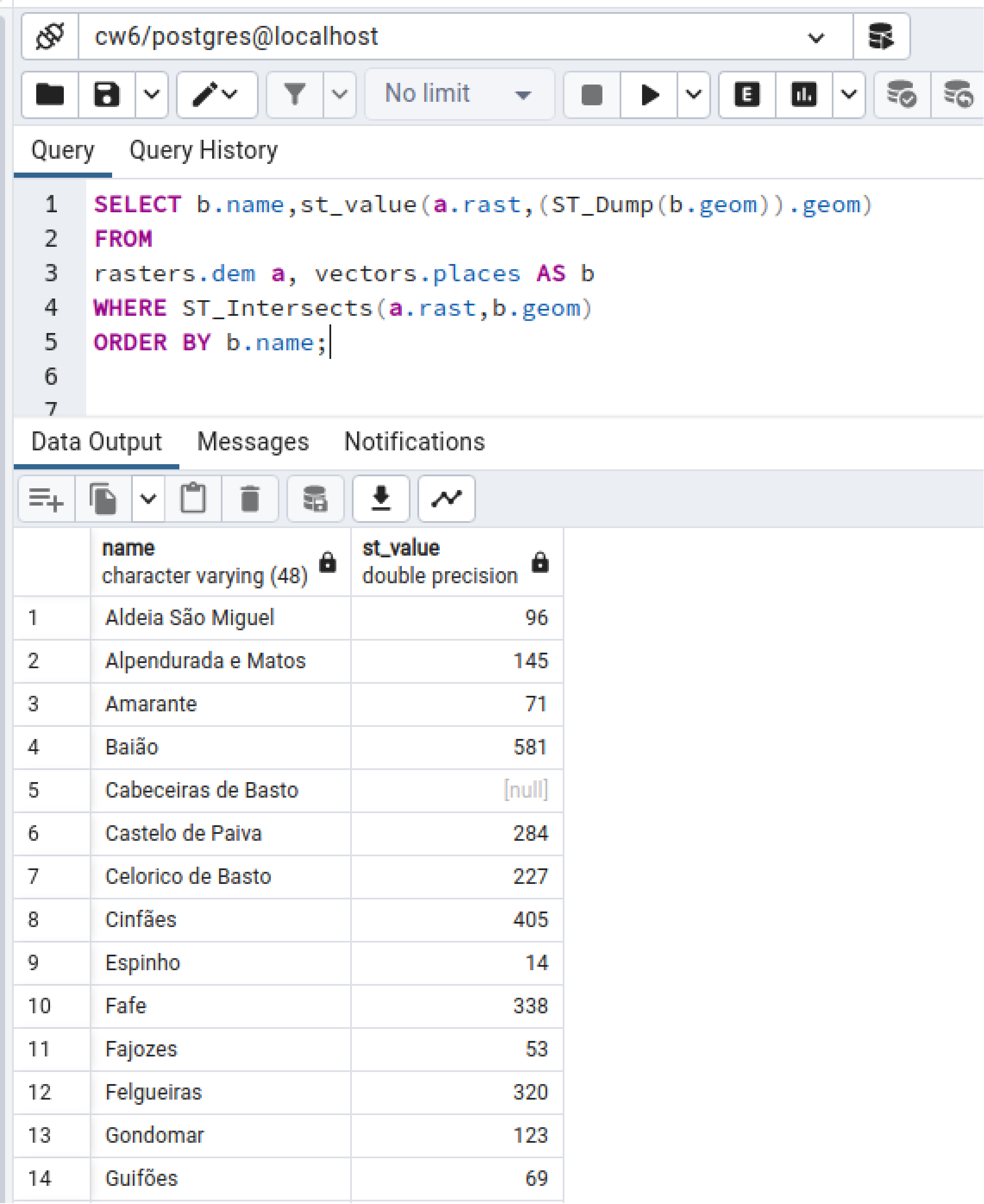
Przykład 8 - ST_SummaryStats w połączeniu z GROUP BY



```sql
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

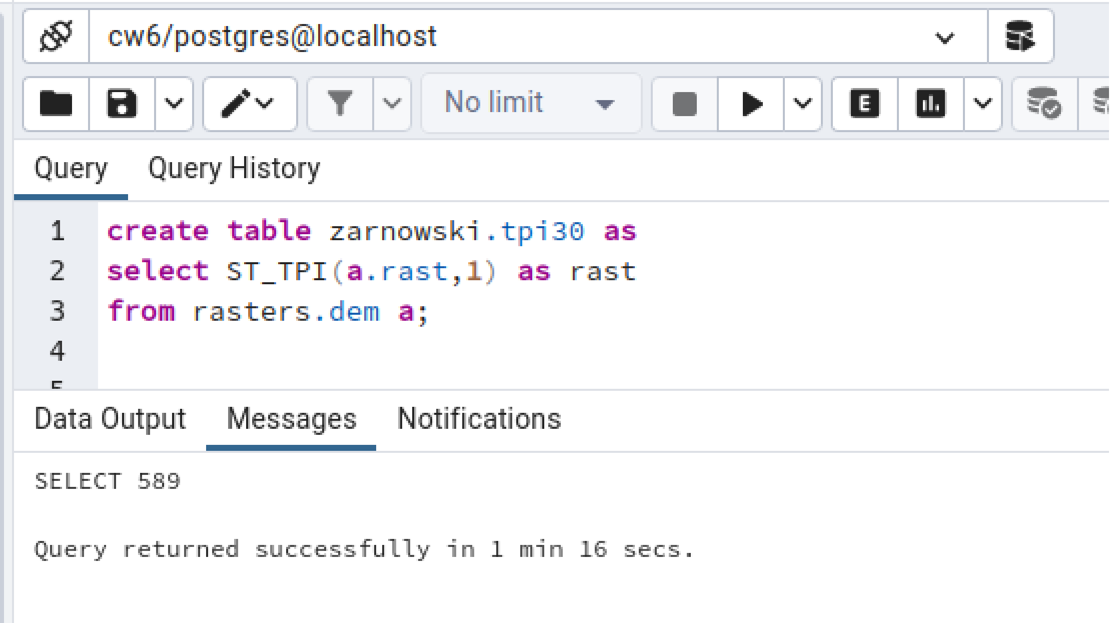
Przykład 9 - ST_Value



```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;
```

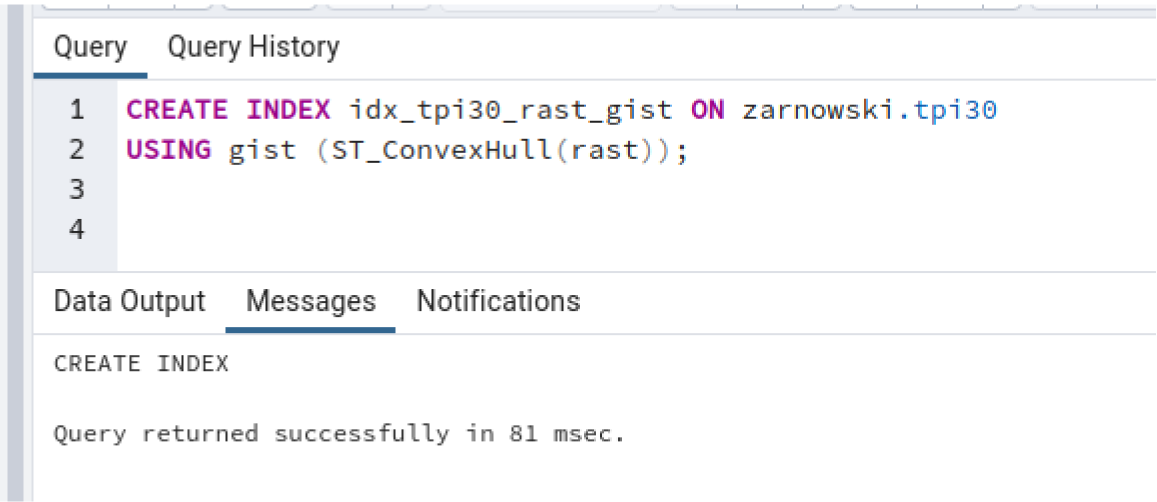
	name character varying (48)	st_value double precision
1	Aldeia São Miguel	96
2	Alpendurada e Matos	145
3	Amarante	71
4	Baião	581
5	Cabeceiras de Basto	[null]
6	Castelo de Paiva	284
7	Celorico de Basto	227
8	Cinfães	405
9	Espinho	14
10	Fafe	338
11	Fajozes	53
12	Felgueiras	320
13	Gondomar	123
14	Guifões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

Przykład 10 - ST_TPI



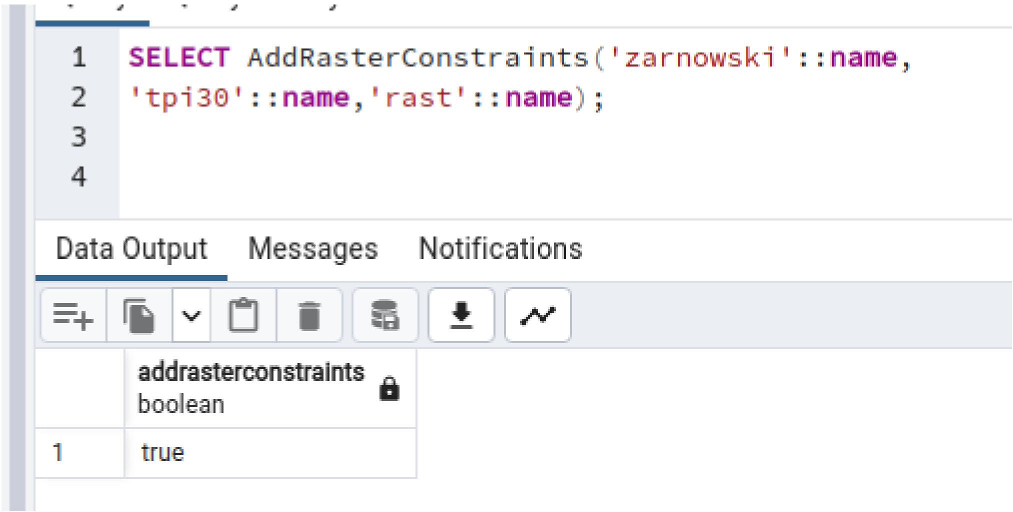
```sql
create table zarnowski.tpi30 as
select ST_TPI(a.rast,1) as rast
from rasters.dem a;
```

```
SELECT 589

Query returned successfully in 1 min 16 secs.
```



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

```
CREATE INDEX

Query returned successfully in 81 msec.
```

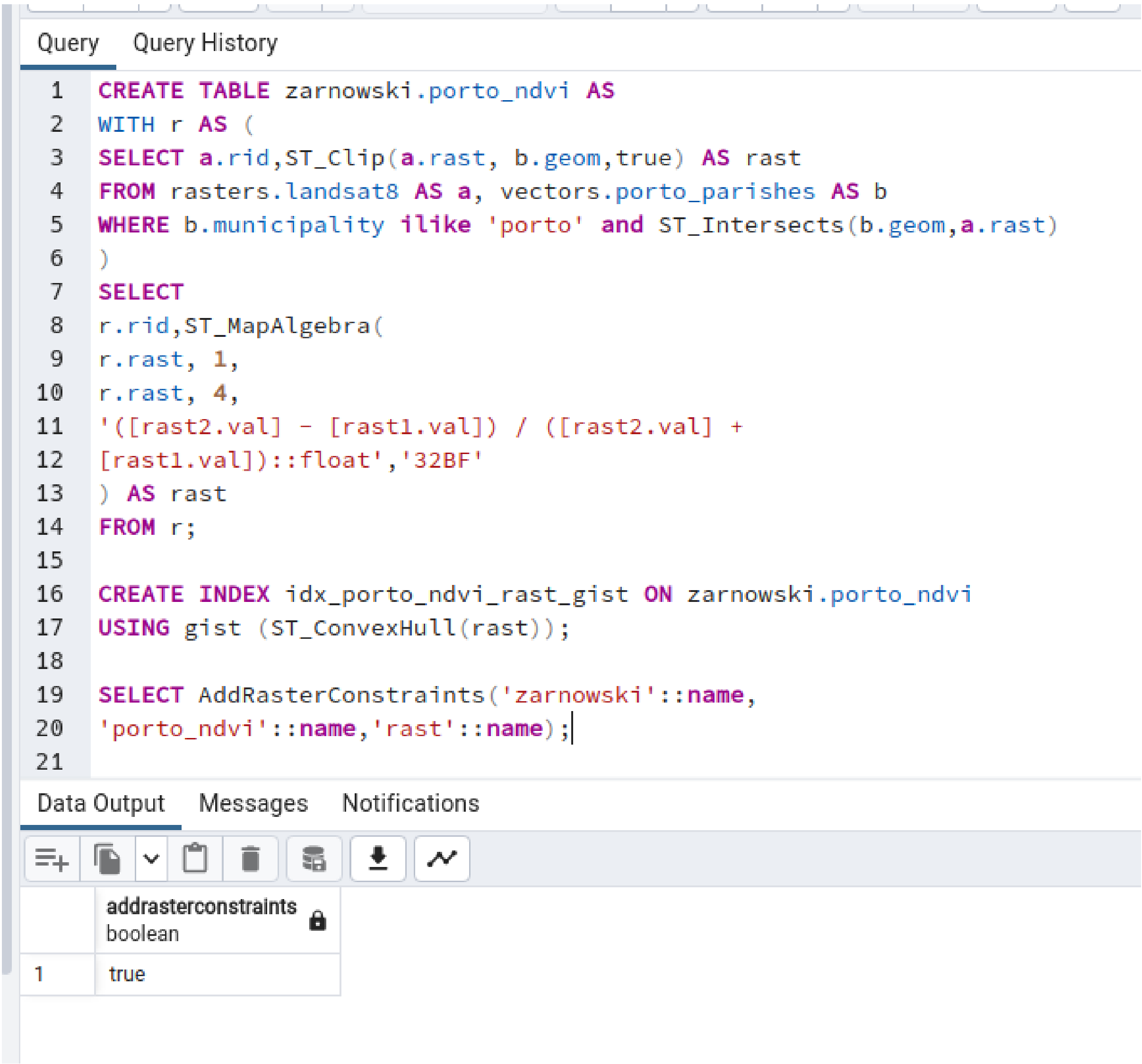


```sql
SELECT AddRasterConstraints('zarnowski'::name,
'tpi30'::name,'rast'::name);
```

	addrasterconstraints boolean
1	true

Algebra map

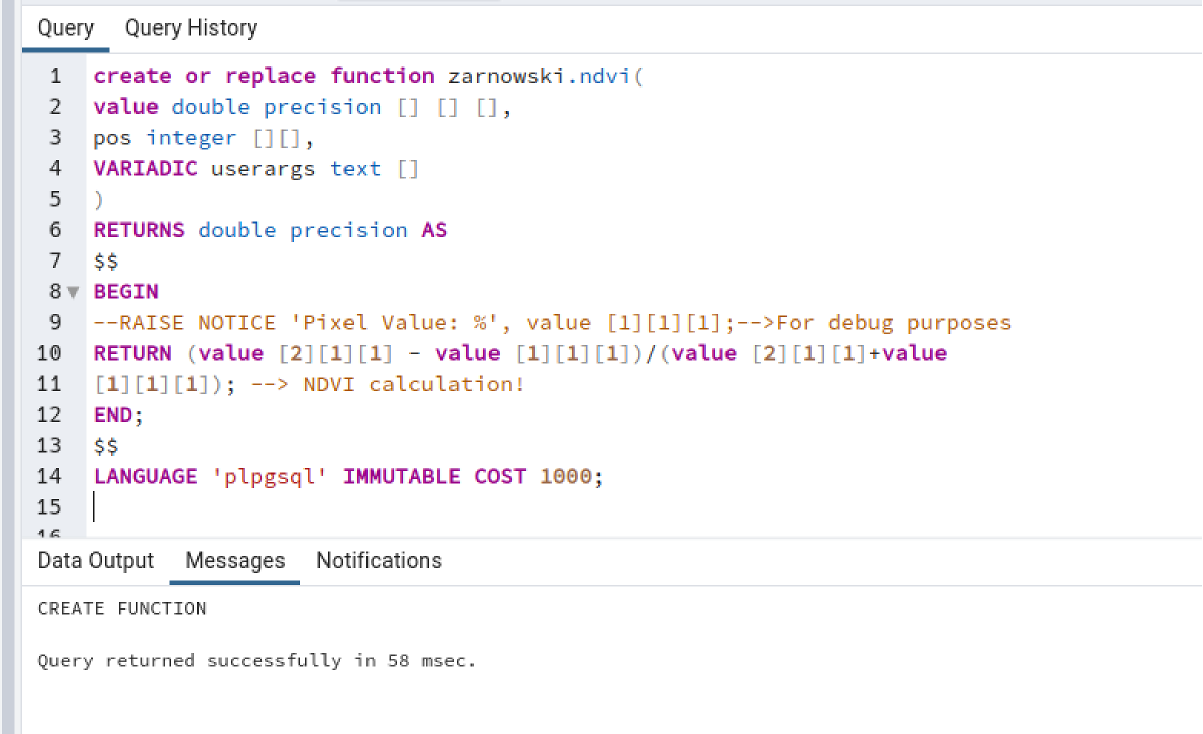
Przykład 1 - Wyrażenie Algebry Map



```sql
CREATE TABLE zarnowski.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 zarnowski.porto_ndvi
USING gist (ST_ConvexHull(rast));

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

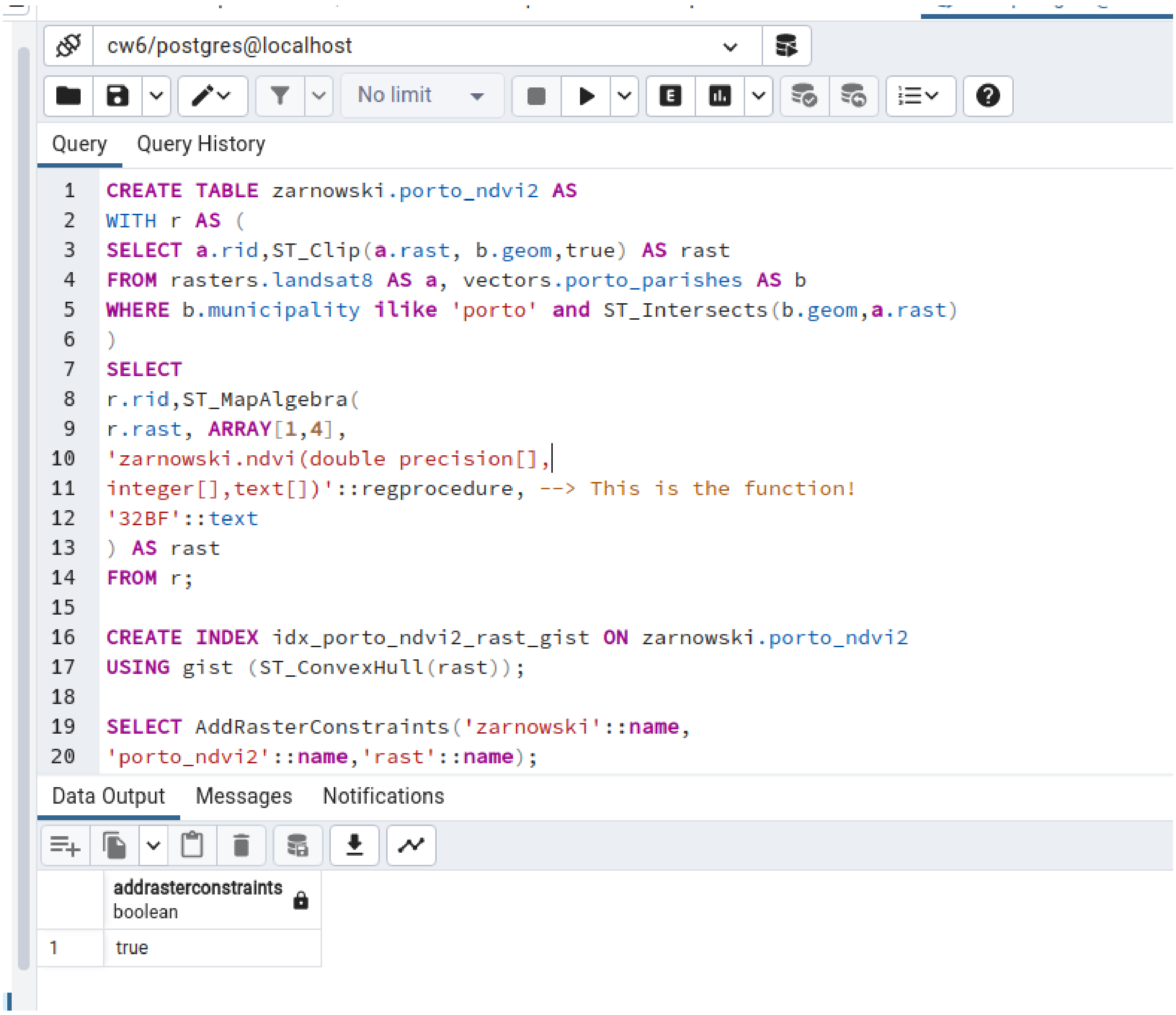
	addrasterconstraints boolean
1	true

Przykład 2 – Funkcja zwrotna



```
create or replace function zarnowski.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 58 msec.
```

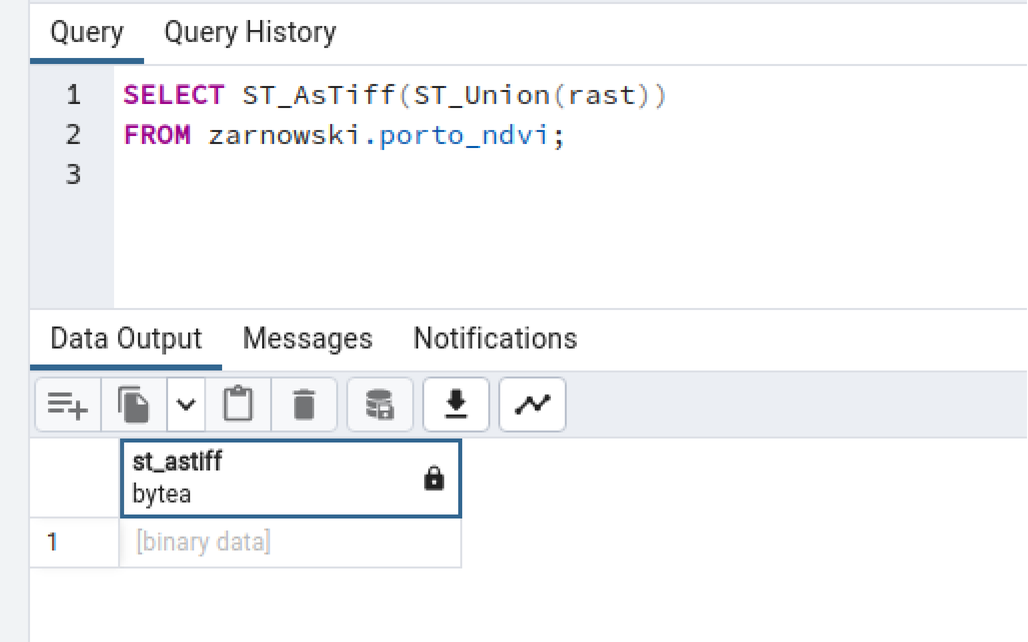


```
CREATE TABLE zarnowski.porto_ndvi2 AS
WITH r AS (
SELECT a.rid,ST_Clip(a.rast, b.geom,true) AS rast
FROM rasters.landsat8 AS a, vectors.porto_parishes AS b
WHERE b.municipality ilike 'porto' and ST_Intersects(b.geom,a.rast)
)
SELECT
r.rid,ST_MapAlgebra(
r.rast, ARRAY[1,4],
'zarnowski.ndvi(double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

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

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

	addrasterconstraints boolean
1	true

Eksport danych

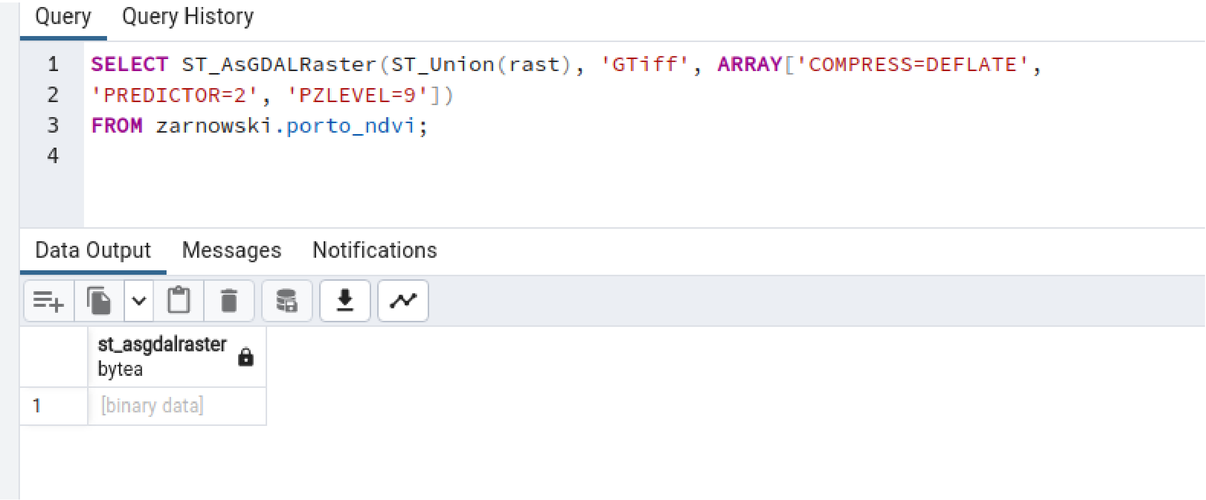
Przykład 1 - ST_AsTiff



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

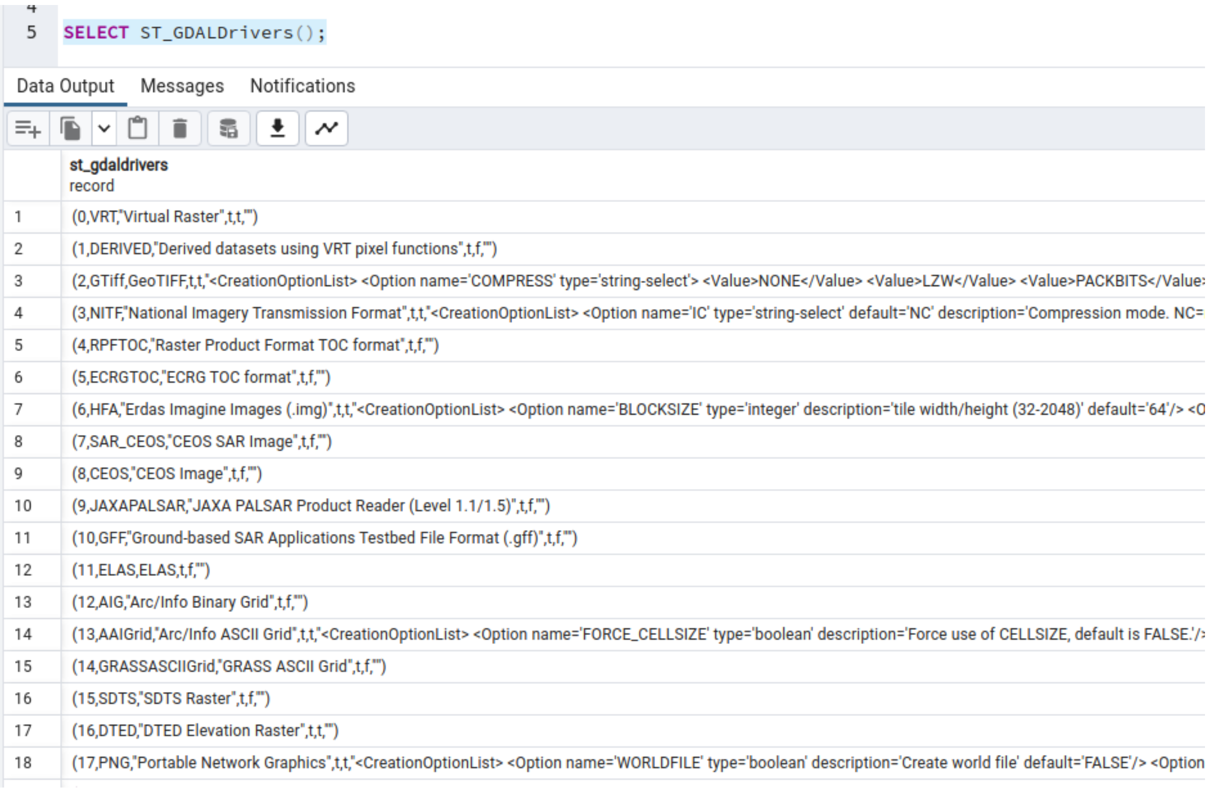
	st_astiff bytea
1	[binary data]

Przykład 2 - ST_AsGDALRaster



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

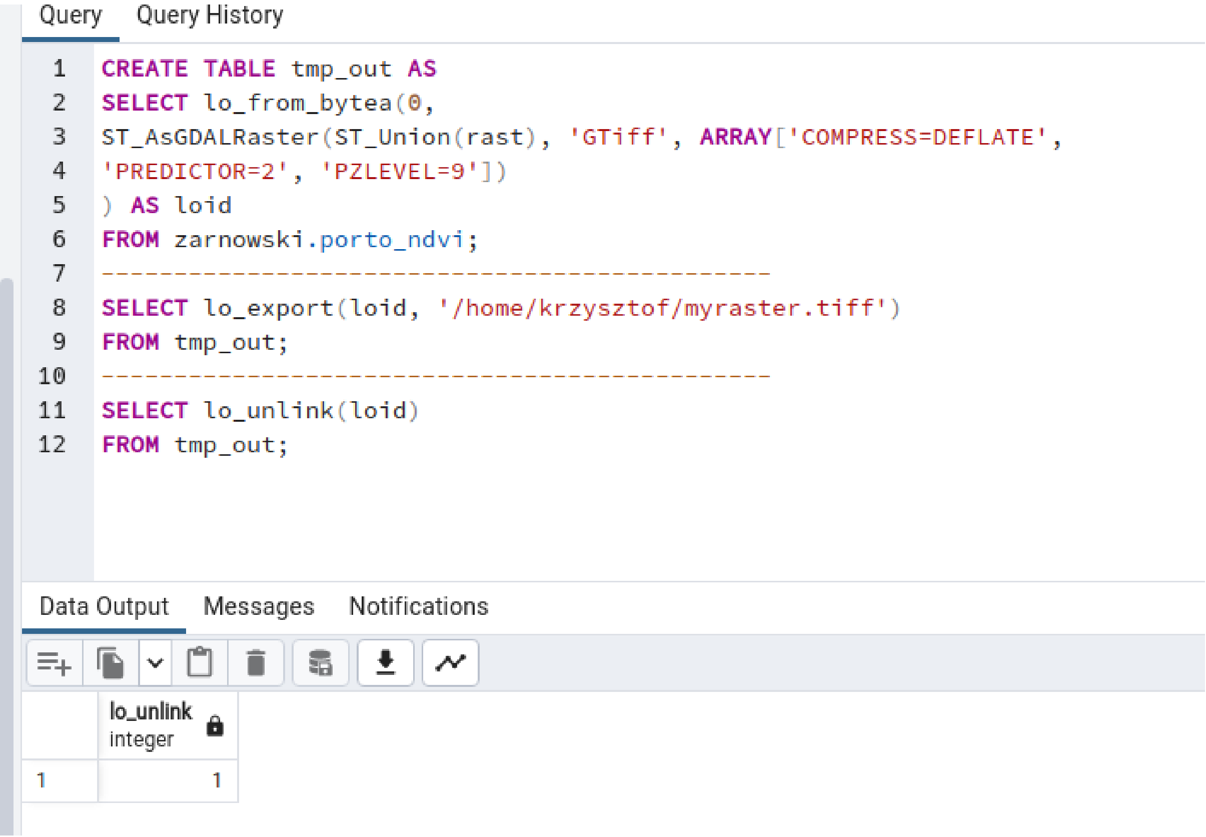
	st_asgdalraster bytea
1	[binary data]



```sql
SELECT ST_GDALDrivers();
```

	st_gdaldrivers record
1	(0,VRT,"Virtual Raster",t,t,"")
2	(1,DERIVED,"Derived datasets using VRT pixel functions",t,f,"")
3	(2,GTiff,GeoTIFF,t,t,"<CreationOptionList> <Option name='COMPRESS' type='string-select'> <Value>NONE</Value> <Value>LZW</Value> <Value>PACKBITS</Value
4	(3,NITF,"National Imagery Transmission Format",t,t,"<CreationOptionList> <Option name='IC' type='string-select' default='NC' description='Compression mode. NC=
5	(4,RPFTOC,"Raster Product Format TOC format",t,f,"")
6	(5,ECRGTOC,"ECRG TOC format",t,f,"")
7	(6,HFA,"Erdas Imagine Images (.img)",t,t,"<CreationOptionList> <Option name='BLOCKSIZE' type='integer' description='tile width/height (32-2048)' default='64'/> <O
8	(7,SAR_CEOS,"CEOS SAR Image",t,f,"")
9	(8,CEOS,"CEOS Image",t,f,"")
10	(9,JAXAPALSAR,"JAXA PALSAR Product Reader (Level 1.1/1.5)",t,f,"")
11	(10,GFF,"Ground-based SAR Applications Testbed File Format (.gff)",t,f,"")
12	(11,ELAS,ELAS,t,f,"")
13	(12,AIG,"Arc/Info Binary Grid",t,f,"")
14	(13,AAIGrid,"Arc/Info ASCII Grid",t,t,"<CreationOptionList> <Option name='FORCE_CELLSIZE' type='boolean' description='Force use of CELLSIZE, default is FALSE.'/
15	(14,GRASSASCIIGrid,"GRASS ASCII Grid",t,f,"")
16	(15,SDTS,"SDTS Raster",t,f,"")
17	(16,DTED,"DTED Elevation Raster",t,t,"")
18	(17,PNG,"Portable Network Graphics",t,t,"<CreationOptionList> <Option name='WORLDFILE' type='boolean' description='Create world file' default='FALSE'/> <Option

Przykład 3 - zapisywanie dużego obiektu

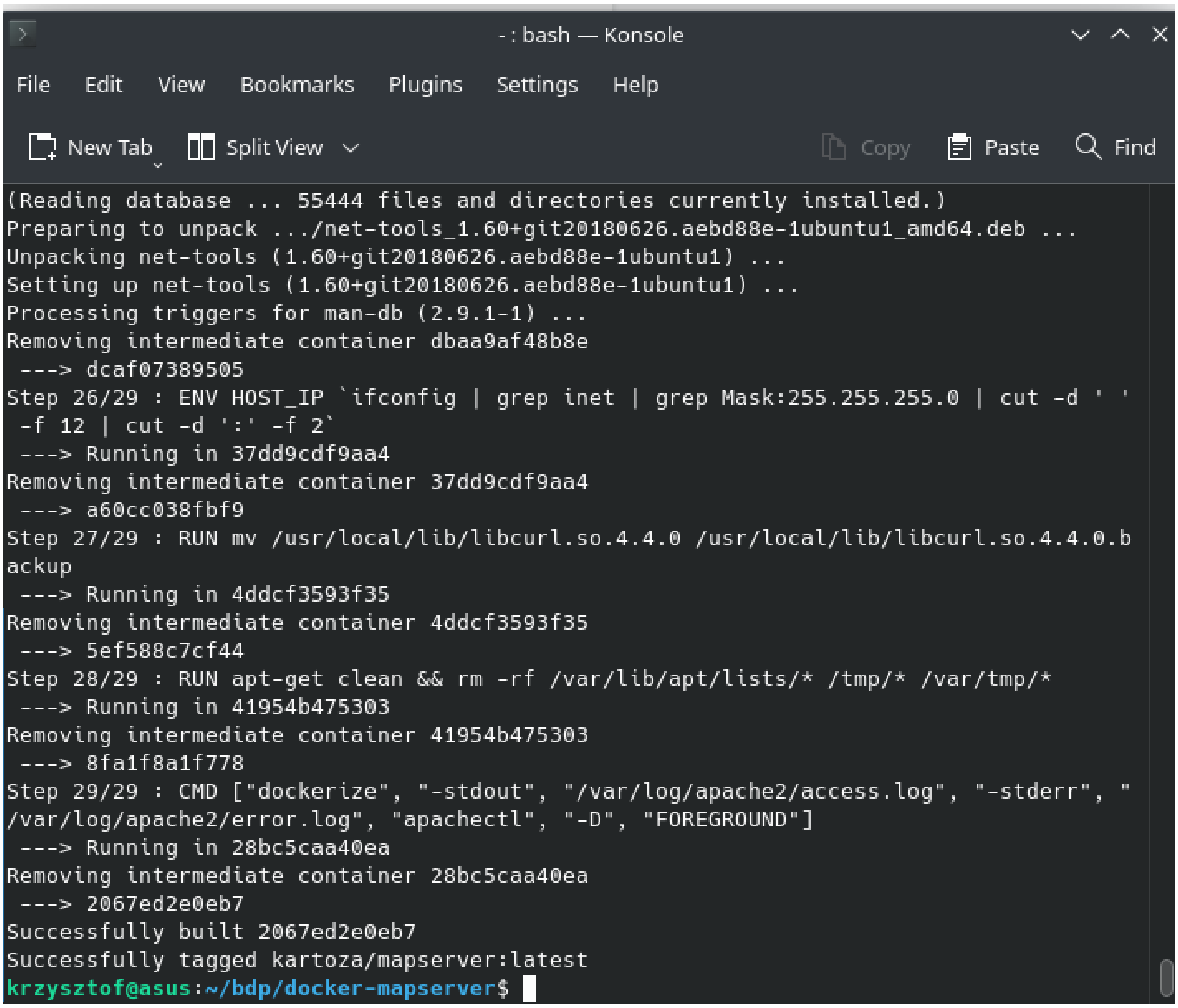


```sql
CREATE TABLE tmp_out AS
SELECT lo_from_bytea(0,
ST_AsGDALRaster(ST_Union(rast), 'GTiff', ARRAY['COMPRESS=DEFLATE',
'PREDICTOR=2', 'PZLEVEL=9'])
) AS loid
FROM zarnowski.porto_ndvi;
----------------------------------------------
SELECT lo_export(loid, '/home/krzysztof/myraster.tiff')
FROM tmp_out;
----------------------------------------------
SELECT lo_unlink(loid)
FROM tmp_out;
```

	lo_unlink integer
1	1

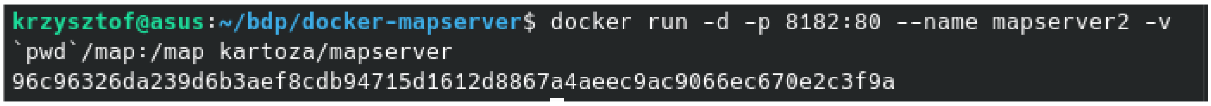
Docker

```
docker build -t kartoza/mapserver .
```

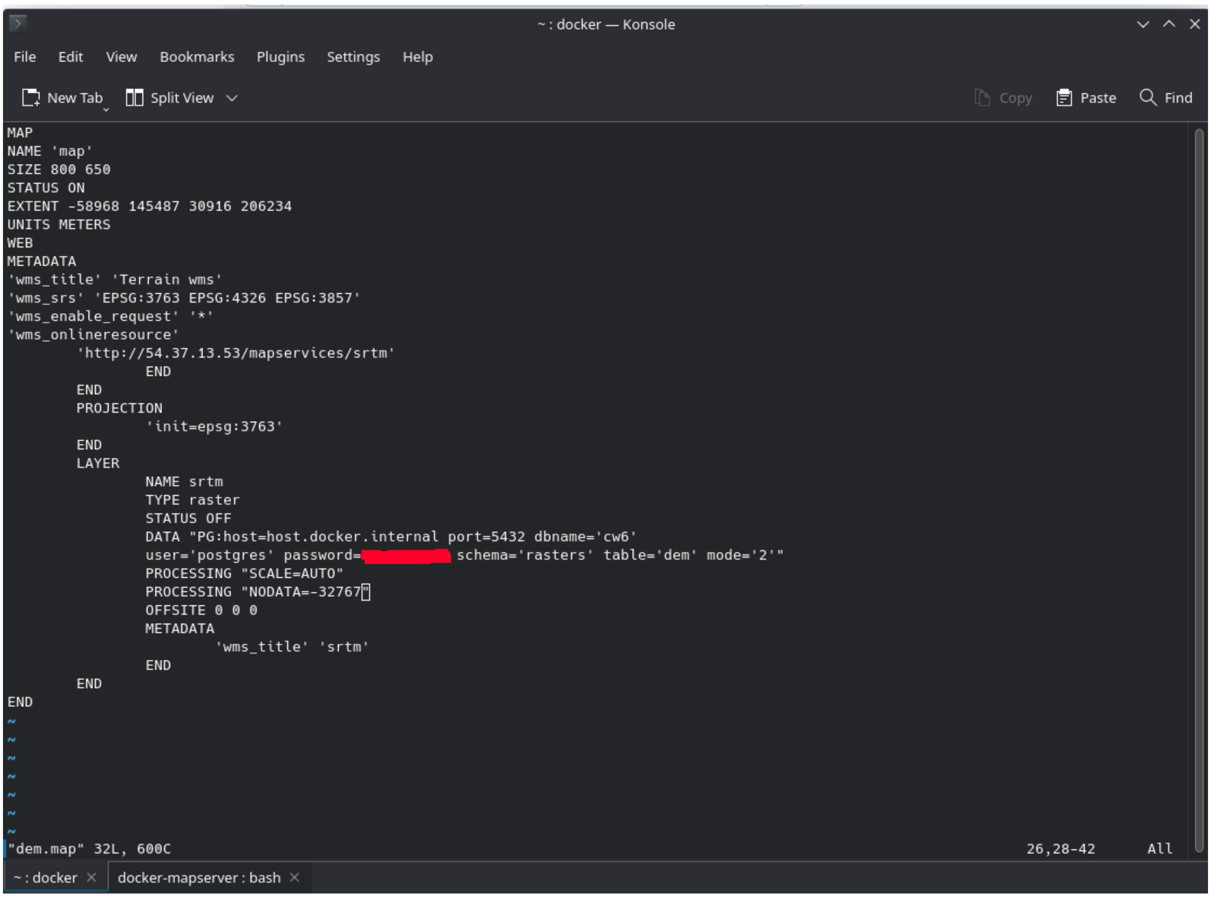


```
(Reading database ... 55444 files and directories currently installed.)
Preparing to unpack .../net-tools_1.60+git20180626.aebd88e-1ubuntu1_amd64.deb ...
Unpacking net-tools (1.60+git20180626.aebd88e-1ubuntu1) ...
Setting up net-tools (1.60+git20180626.aebd88e-1ubuntu1) ...
Processing triggers for man-db (2.9.1-1) ...
Removing intermediate container dbaa9af48b8e
 ---> dcaf07389505
Step 26/29 : ENV HOST_IP `ifconfig | grep inet | grep Mask:255.255.255.0 | cut -d ' '
 -f 12 | cut -d ':' -f 2`
 ---> Running in 37dd9cdf9aa4
Removing intermediate container 37dd9cdf9aa4
 ---> a60cc038fbf9
Step 27/29 : RUN mv /usr/local/lib/libcurl.so.4.4.0 /usr/local/lib/libcurl.so.4.4.0.b
ackup
 ---> Running in 4ddcf3593f35
Removing intermediate container 4ddcf3593f35
 ---> 5ef588c7cf44
Step 28/29 : RUN apt-get clean && rm -rf /var/lib/apt/lists/* /tmp/* /var/tmp/*
 ---> Running in 41954b475303
Removing intermediate container 41954b475303
 ---> 8fa1f8a1f778
Step 29/29 : CMD ["dockerize", "-stdout", "/var/log/apache2/access.log", "-stderr", "
/var/log/apache2/error.log", "apachectl", "-D", "FOREGROUND"]
 ---> Running in 28bc5caa40ea
Removing intermediate container 28bc5caa40ea
 ---> 2067ed2e0eb7
Successfully built 2067ed2e0eb7
Successfully tagged kartoza/mapserver:latest
krzysztof@asus:~/bdp/docker-mapserver$
```

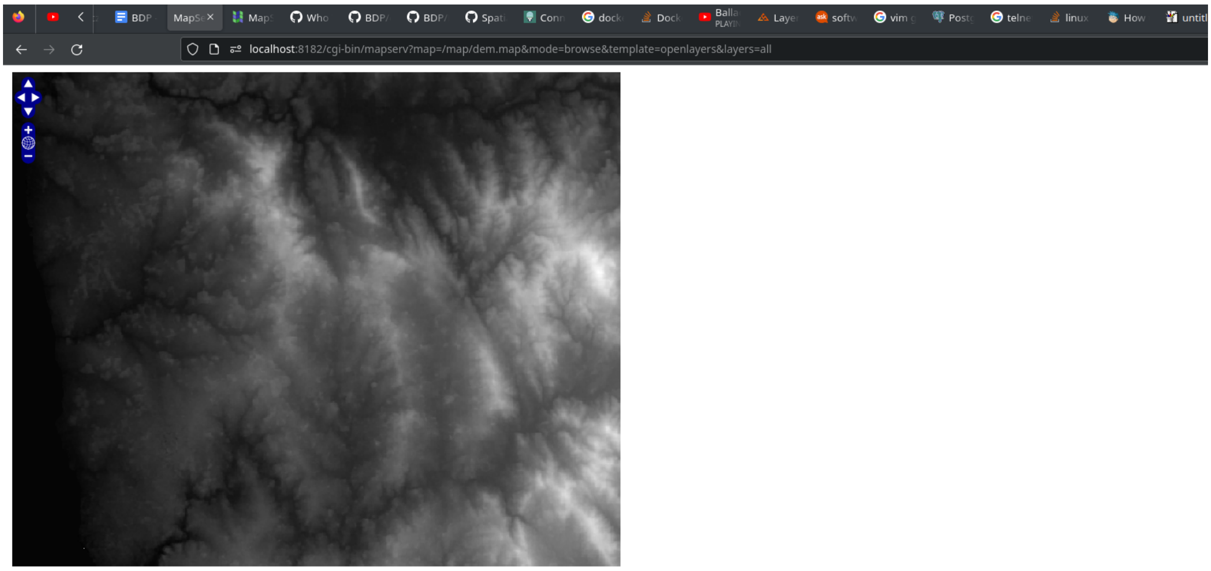
```
docker run -d -p 8182:80 --name mapserver2 -v `pwd`/map:/map
kartoza/mapserver
```



```
krzysztof@asus:~/bdp/docker-mapserver$ docker run -d -p 8182:80 --name mapserver2 -v
`pwd`/map:/map kartoza/mapserver
96c96326da239d6b3aef8cdb94715d1612d8867a4aeec9ac9066ec670e2c3f9a
```



```
MAP
NAME 'map'
SIZE 800 650
STATUS ON
EXTENT -58968 145487 30916 206234
UNITS METERS
WEB
METADATA
'wms_title' 'Terrain wms'
'wms_srs' 'EPSG:3763 EPSG:4326 EPSG:3857'
'wms_enable_request' '*'
'wms_onlineresource'
        'http://54.37.13.53/mapservices/srtm'
                END
        END
        PROJECTION
                'init=epsg:3763'
        END
        LAYER
                NAME srtm
                TYPE raster
                STATUS OFF
                DATA "PG:host=host.docker.internal port=5432 dbname='cw6'
                user='postgres' password=[redacted] schema='rasters' table='dem' mode='2'"
                PROCESSING "SCALE=AUTO"
                PROCESSING "NODATA=-32767"
                OFFSITE 0 0 0
                METADATA
                        'wms_title' 'srtm'
                END
        END
END
~
~
~
~
~
~
~
"dem.map" 32L, 600C                                                     26,28-42      All
```



localhost:8182/cgi-bin/mapserv?map=/map/dem.map&mode=browse&template=openlayers&layers=all