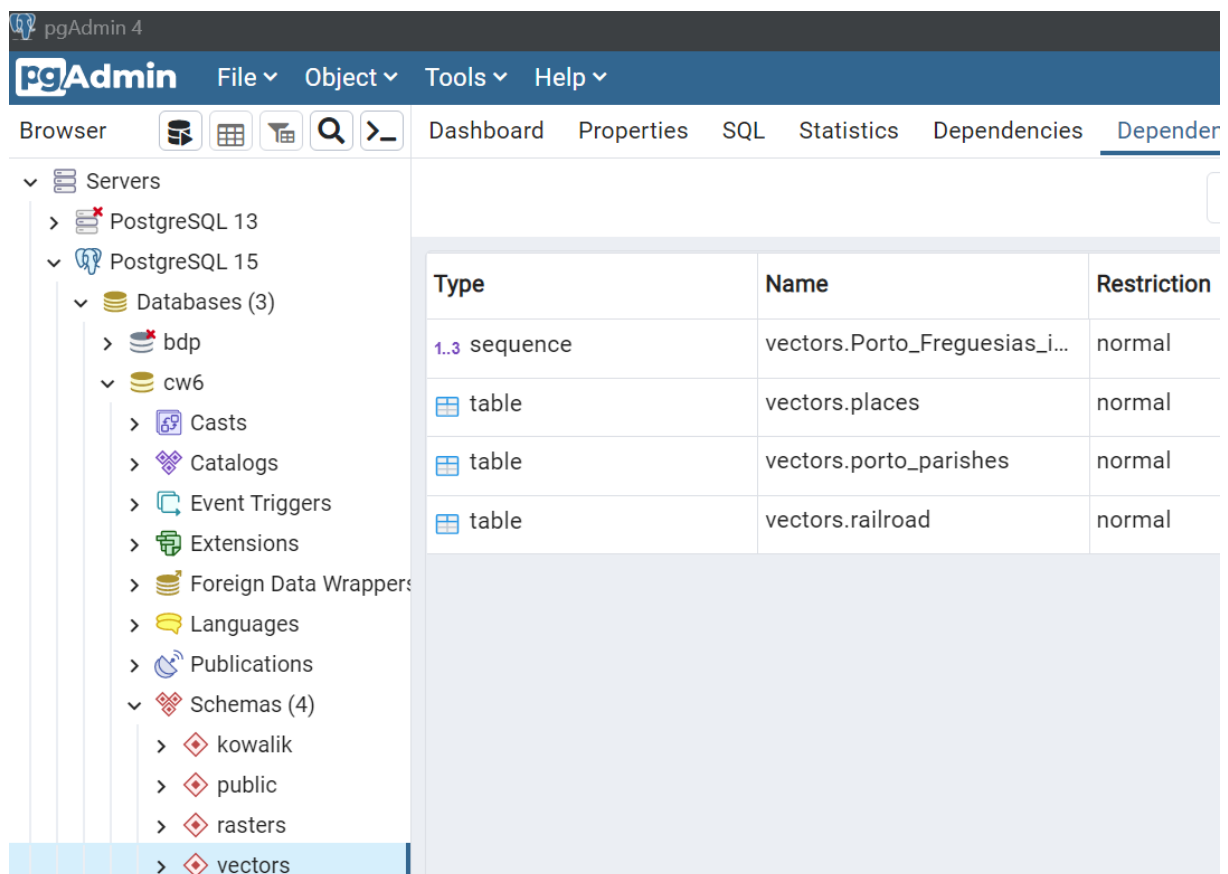


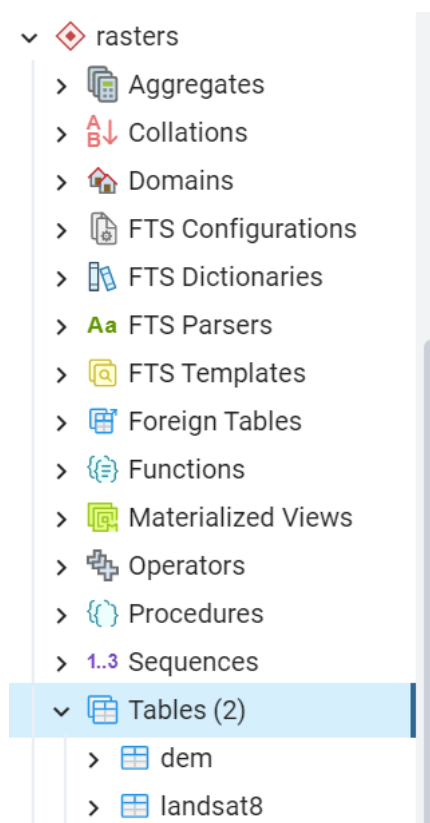
nowa baza danych i struktura bazy danych



The screenshot shows the pgAdmin 4 interface. The left sidebar displays a tree view of the database structure. The 'vectors' schema is selected, showing its contents. The main pane on the right displays a table with the following data:

Type	Name	Restriction
1..3 sequence	vectors.Porto_Freguesias_i...	normal
table	vectors.places	normal
table	vectors.porto_parishes	normal
table	vectors.railroad	normal

ładowanie danych rastrowych



The screenshot shows the pgAdmin 4 interface with the 'rasters' schema selected in the left sidebar. The 'Tables (2)' folder is expanded, showing two tables: 'dem' and 'landsat8'.

Tworzenie rastrów z istniejących rastrów i interakcja z wektorami

Przykład 1 - ST_Intersects

The screenshot shows the PostgreSQL GUI interface. On the left, the 'kowalik' database is selected, and the 'intersects' table is highlighted under 'Tables (1)'. The main query editor displays the following SQL code:

```
1 CREATE TABLE kowalik.intersects AS
2 SELECT a.rast, b.municipality
3 FROM rasters.dem AS a, vectors.porto_parishes AS b
4 WHERE ST_Intersects(a.rast, b.geom) AND b.municipality ilike 'porto';
5
```

The 'Query' tab is active, and the 'Data Output' tab shows 'SELECT 25'. The 'Messages' tab indicates 'Query returned successfully in 107 msec.'

This block contains three screenshots showing the continuation of the database setup:

- Left Screenshot:** The query editor shows the command to alter the table and add a primary key:

```
1 alter table kowalik.intersects
2 add column rid SERIAL PRIMARY KEY;
```

The 'Data Output' tab shows 'ALTER TABLE' and the 'Messages' tab shows 'Query returned successfully in 52 msec.'
- Middle Screenshot:** The query editor shows the command to create an index:

```
1 CREATE INDEX idx_intersects_rast_gist ON kowalik.intersects
2 USING gist (ST_ConvexHull(rast));
3
```

The 'Data Output' tab shows 'CREATE INDEX' and the 'Messages' tab shows 'Query returned successfully in 49 msec.'
- Right Screenshot:** The query editor shows the command to add raster constraints:

```
1 SELECT AddRasterConstraints('kowalik'::name,
2                             'intersects'::name, 'rast'::name);
```

The 'Data Output' tab shows the results of the command:

addrasterconstraints	boolean
1	true

Przykład 2 - ST_Clip

The screenshot shows the PostgreSQL GUI interface. On the left, the 'kowalik' database is selected, and the 'clip' table is highlighted under 'Tables (2)'. The main query editor displays the following SQL code:

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

The 'Query' tab is active, and the 'Data Output' tab shows 'SELECT 25'. The 'Messages' tab indicates 'Query returned successfully in 80 msec.'

Przykład 3 - ST_Union

The screenshot shows the PostgreSQL GUI interface. On the left, the 'kowalik' database is selected, and the 'Tables (3)' folder is expanded, showing 'clip', 'intersects', and 'union'. The 'Query' tab is active, displaying the following SQL query:

```
1 CREATE TABLE kowalik.union AS
2 SELECT ST_Union(ST_Clip(a.rast, b.geom, true))
3 FROM rasters.dem AS a, vectors.porto_parishes AS b
4 WHERE b.municipality ilike 'porto' and ST_Intersects(b.geom,a.rast);
5
```

The 'Messages' tab is selected, showing the message: 'Query returned successfully in 235 msec.'

Tworzenie rastrów z wektorów (rastrowanie)

Przykład 1 - ST_AsRaster

The screenshot shows the PostgreSQL GUI interface. On the left, the 'kowalik' database is selected, and the 'Tables (4)' folder is expanded, showing 'clip', 'intersects', 'porto_parishes', and 'union'. The 'Query' tab is active, displaying the following SQL query:

```
1 CREATE TABLE kowalik.porto_parishes AS WITH r AS (
2 SELECT rast FROM rasters.dem
3 LIMIT 1)
4
5 SELECT ST_AsRaster(a.geom,r.rast,'8BUI',a.id,-32767) AS rast
6 FROM vectors.porto_parishes AS a, r
7 WHERE a.municipality ilike 'porto';
8
9
```

The 'Messages' tab is selected, showing the message: 'Query returned successfully in 86 msec.'

Przykład 2 - ST_Union

The screenshot shows the PostgreSQL GUI interface. On the left, the 'kowalik' database is selected, and the 'Tables (4)' folder is expanded, showing 'clip', 'intersects', 'porto_parishes', and 'union'. The 'Query' tab is active, displaying the following SQL query:

```
1 DROP TABLE kowalik.porto_parishes; --> drop table porto_parishes first
2 CREATE TABLE kowalik.porto_parishes AS
3 WITH r AS (
4 SELECT rast FROM rasters.dem
5 LIMIT 1
6 )
7 SELECT st_union(ST_AsRaster(a.geom,r.rast,'8BUI',a.id,-32767)) AS rast
8 FROM vectors.porto_parishes AS a, r
9 WHERE a.municipality ilike 'porto';
10
```

The 'Messages' tab is selected, showing the message: 'Query returned successfully in 86 msec.'

Przykład 3 - ST_Tile

The screenshot shows the PostgreSQL GUI interface. On the left, the database schema 'kowalik' is expanded, showing various database objects. The main query editor displays the following SQL code:

```
1 DROP TABLE kowalik.porto_parishes; --> drop table porto_parishes first
2 CREATE TABLE kowalik.porto_parishes AS
3 WITH r AS (
4 SELECT rast FROM rasters.dem
5 LIMIT 1 )
6 SELECT st_tile(st_union(ST_AsRaster(a.geom,r.rast,'8BUI',a.id,-
7 32767)),128,128,true,-32767) AS rast
8 FROM vectors.porto_parishes AS a, r
9 WHERE a.municipality ilike 'porto';
10
```

Below the query editor, the 'Messages' tab is active, showing the message: 'Query returned successfully in 110 msec.'

Konwertowanie rastrow na wektory (wektoryzowanie)

Przykład 1 - ST_Intersection

The screenshot shows the PostgreSQL GUI interface. On the left, the database schema 'kowalik' is expanded, showing various database objects. The main query editor displays the following SQL code:

```
1 create table kowalik.intersection as
2 SELECT
3 a.rid,(ST_Intersection(b.geom,a.rast)).geom,(ST_Intersection(b.geom,a.rast)
4 ).val
5 FROM rasters.landsat8 AS a, vectors.porto_parishes AS b
6 WHERE b.parish ilike 'paranhos' and ST_Intersects(b.geom,a.rast);
7
```

Below the query editor, the 'Messages' tab is active, showing the message: 'Query returned successfully in 2 secs 871 msec.'

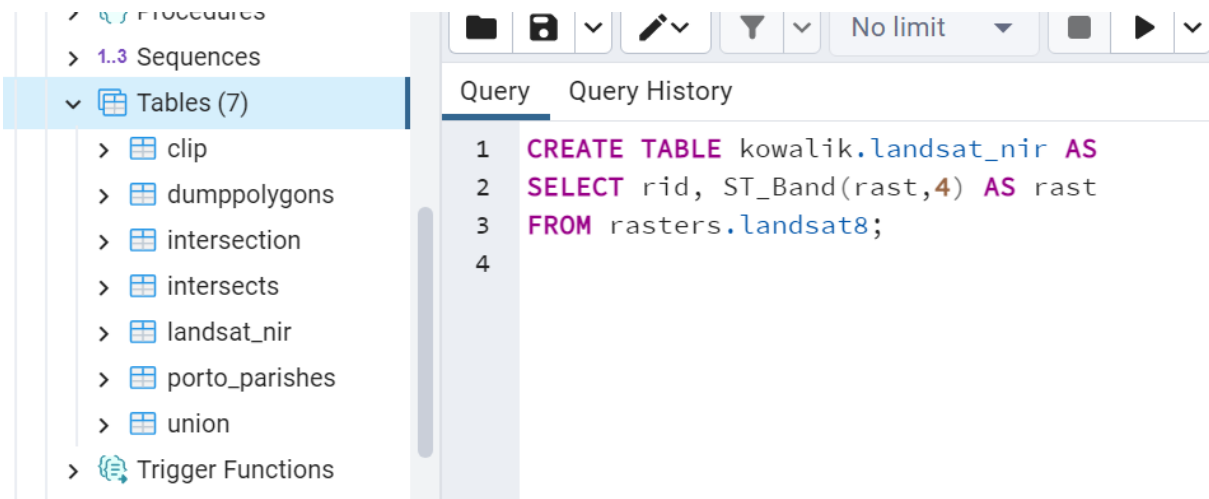
Przykład 2 - ST_DumpAsPolygons

The screenshot shows the PostgreSQL GUI interface. On the left, the database schema 'kowalik' is expanded, showing various database objects. The main query editor displays the following SQL code:

```
1 CREATE TABLE kowalik.dumppolygons AS
2 SELECT
3 a.rid,(ST_DumpAsPolygons(ST_Clip(a.rast,b.geom)).geom,
4 (ST_DumpAsPolygons(ST_Clip(a.rast,b.geom)).val
5 FROM rasters.landsat8 AS a, vectors.porto_parishes AS b
6 WHERE b.parish ilike 'paranhos' and ST_Intersects(b.geom,a.rast);
```

Analiza rastrów


Przykład 1 - ST_Band



The screenshot shows a GIS software interface with a left sidebar containing a tree view of database objects. The 'Tables (7)' folder is expanded, showing a list of tables: clip, dumppolygons, intersection, intersects, landsat_nir, porto_parishes, and union. The 'Query' tab is active in the main window, displaying a SQL query. The query is as follows:

```
1 CREATE TABLE kowalik.landsat_nir AS
2 SELECT rid, ST_Band(rast,4) AS rast
3 FROM rasters.landsat8;
4
```

Przykład 2 - ST_Clip



The screenshot shows a GIS software interface with a left sidebar containing a tree view of database objects. The 'Tables (8)' folder is expanded, showing a list of tables: clip, dumppolygons, intersection, intersects, landsat_nir, paranhos_dem, porto_parishes, and union. The 'Query' tab is active in the main window, displaying a SQL query. The query is as follows:

```
1 CREATE TABLE kowalik.paranhos_dem AS
2 SELECT a.rid,ST_Clip(a.rast, b.geom,true) as rast
3 FROM rasters.dem AS a, vectors.porto_parishes AS b
4 WHERE b.parish ilike 'paranhos' and ST_Intersects(b.geom,a.rast);
5
6
```

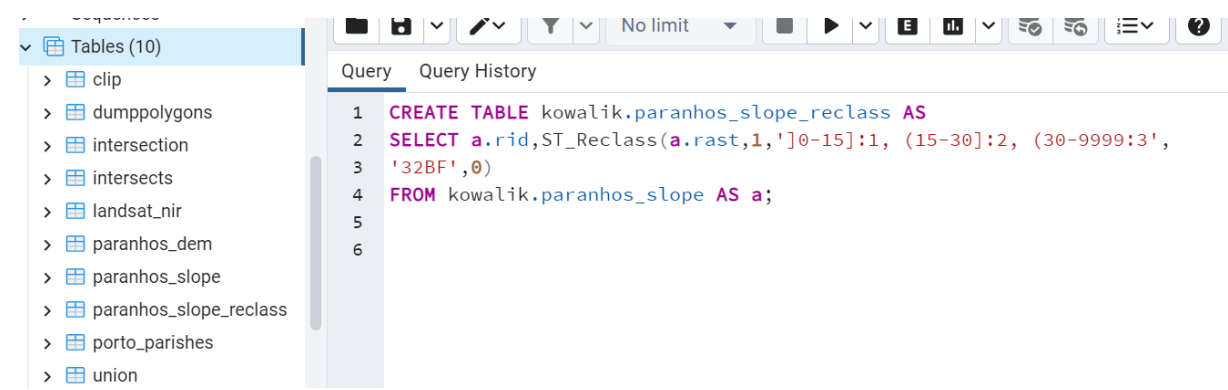
Przykład 3 - ST_Slope



The screenshot shows a GIS software interface with a left sidebar containing a tree view of database objects. The 'Tables (9)' folder is expanded, showing a list of tables: clip, dumppolygons, intersection, intersects, landsat_nir, paranhos_dem, and paranhos_slope. The 'Query' tab is active in the main window, displaying a SQL query. The query is as follows:

```
1 CREATE TABLE kowalik.paranhos_slope AS
2 SELECT a.rid,ST_Slope(a.rast,1,'32BF','PERCENTAGE') as rast
3 FROM kowalik.paranhos_dem AS a;
4
5
```

Przykład 4 - ST_Reclass



The screenshot shows a GIS software interface with a left sidebar containing a tree view of database objects. The 'Tables (10)' folder is expanded, showing a list of tables: clip, dumppolygons, intersection, intersects, landsat_nir, paranhos_dem, paranhos_slope, paranhos_slope_reclass, porto_parishes, and union. The 'Query' tab is active in the main window, displaying a SQL query. The query is as follows:

```
1 CREATE TABLE kowalik.paranhos_slope_reclass AS
2 SELECT a.rid,ST_Reclass(a.rast,1,']0-15]:1, (15-30]:2, (30-9999:3',
3 '32BF',0)
4 FROM kowalik.paranhos_slope AS a;
5
6
```

Przykład 5 - ST_SummaryStats

Query

Query History

1

SELECT st_summarystats(a.rast) AS stats

2

FROM kowalik.paranhos_dem AS a;

3









4


|

Data Output

Messages

Notifications



	stats	
	summarystats	
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)	

Przykład 6 - ST_SummaryStats oraz Union

Query

Query History

1

SELECT st_summarystats(ST_Union(a.rast))









2


FROM kowalik.paranhos_dem AS a;

Data Output

Messages

Notifications



	st_summarystats	
	summarystats	
1	(9977,1222455,122.52731281948482,16.908004202736272,87,158)	

Przykład 7 - ST_SummaryStats z lepszą kontrolą złożonego typu danych

Query

Query History

```
1 WITH t AS (  
2 SELECT st_summarystats(ST_Union(a.rast)) AS stats  
3 FROM kowalik.paranhos_dem AS a  
4 )  
5 SELECT (stats).min,(stats).max,(stats).mean FROM t;
```

Data Output

Messages

Notifications

</

Przykład 8 - ST_SummaryStats w połączeniu z GROUP BY

Query

Query History

1

WITH t AS (

2

SELECT b.parish AS parish, st_summarystats(ST_Union(ST_Clip(a.rast,

3

b.geom,true))) AS stats

4

FROM rasters.dem AS a, vectors.porto_parishes AS b

5

WHERE b.municipality ilike 'porto' and ST_Intersects(b.geom,a.rast)

6

group by b.parish

7

)

8

SELECT parish,(stats).min,(stats).max,(stats).mean FROM t;

Data Output

Messages

Notifications

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

Przykład 9 - ST_Value

Query Query History

```
1 SELECT b.name,st_value(a.rast,(ST_Dump(b.geom)).geom)
2 FROM
3 rasters.dem a, vectors.places AS b
4 WHERE ST_Intersects(a.rast,b.geom)
5 ORDER BY b.name;
6
```

Data Output Messages Notifications



	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

Total rows: 6 of 62 Query complete: 00:00:00.074

Przykład 10 - ST_TPI

> clip

> dumppolygons

> intersection

> intersects

> landsat_nir

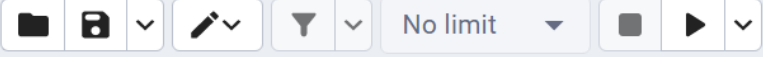
> paranhos_dem

> paranhos_slope

> paranhos_slope_reclass

> porto_parishes

> tpi30



Query Query History

```
1 create table kowalik.tpi30 as
2 select ST_TPI(a.rast,1) as rast
3 from rasters.dem a;
```

1 CREATE INDEX idx_tpi30_rast_gist ON kowalik.tpi30
2 USING gist (ST_ConvexHull(rast));
3

Data Output Messages Notifications

CREATE INDEX

Query returned successfully in 59 msec.

Query Query History

```

1 SELECT AddRasterConstraints('kowalik'::name,
2 'tpi30'::name,'rast'::name);
3

```

Data Output Messages Notifications

	addrasterconstraints boolean	
1	true	

Algebra map

Przykład 1 - Wyrażenie Algebra Map

Tables (12)

clip

dumppolygons

intersection

intersects

landsat_nir

paranhos_dem

paranhos_slope

paranhos_slope_reclass

porto_ndvi

porto_parishes

tpi30

union

Trigger Functions

Tunes

Query Query History

```

1 CREATE TABLE kowalik.porto_ndvi AS
2 WITH r AS (
3 SELECT a.rid,ST_Clip(a.rast, b.geom,true) AS rast
4 FROM rasters.landsat8 AS a, vectors.porto_parishes AS b
5 WHERE b.municipality ilike 'porto' and ST_Intersects(b.geom,a.rast)
6 )
7 SELECT
8 r.rid,ST_MapAlgebra(
9 r.rast, 1,
10 r.rast, 4,
11 '([rast2.val] - [rast1.val]) / ([rast2.val] +
12 [rast1.val])::float','32BF'
13 ) AS rast
14 FROM r;

```

Query Query History

```

1 CREATE INDEX idx_porto_ndvi_rast_gist ON kowalik.porto_ndvi
2 USING gist (ST_ConvexHull(rast));
3

```

Data Output Messages Notifications

CREATE INDEX

Query returned successfully in 48 msec.

Query Query History

```

1 SELECT AddRasterConstraints('kowalik'::name,
2 'porto_ndvi'::name,'rast'::name);
3

```

Data Output Messages Notifications

	addrasterconstraints boolean	
1	true	

Przykład 2 – Funkcja zwrotna

Query Query History

```
1 create or replace function kowalik.ndvi(  
2 value double precision [] [] [],  
3 pos integer [][],  
4 VARIADIC userargs text []  
5 )  
6 RETURNS double precision AS  
7 $$  
8 BEGIN  
9 --RAISE NOTICE 'Pixel Value: %', value [1][1][1];-->For debug purposes  
10 RETURN (value [2][1][1] - value [1][1][1])/(value [2][1][1]+value  
11 [1][1][1]); --> NDVI calculation!  
12 END;  
13 $$  
14 LANGUAGE 'plpgsql' IMMUTABLE COST 1000;
```

Data Output Messages Notifications

CREATE FUNCTION

Query returned successfully in 56 msec.

landsat_nir

paranhos_dem

paranhos_slope

paranhos_slope_reclass

porto_ndvi

porto_ndvi2

porto_parishes

tpi30

union

Trigger Functions

Types

Views

public

Aggregates

Collations

Domains

FTS Configurations

Query Query History

1 CREATE TABLE kowalik.porto_ndvi2 AS
2 WITH r AS (
3 SELECT a.rid,ST_Clip(a.rast, b.geom,true) AS rast
4 FROM rasters.landsat8 AS a, vectors.porto_parishes AS b
5 WHERE b.municipality ilike 'porto' and ST_Intersects(b.geom,a.rast)
6)
7 SELECT
8 r.rid,ST_MapAlgebra(
9 r.rast, ARRAY[1,4],
10 'kowalik.ndvi(double precision[],
11 integer[],text[])::regprocedure,
12 '32BF'::text
13) AS rast
14 FROM r;

Data Output Messages Notifications

SELECT 29

Query returned successfully in 203 msec.

Query Query History

```
1 CREATE INDEX idx_porto_ndvi2_rast_gist ON kowalik.porto_ndvi2  
2 USING gist (ST_ConvexHull(rast));
```

Data Output Messages Notifications

CREATE INDEX

Query returned successfully in 57 msec.

Query Query History

```
1 SELECT AddRasterConstraints('kowalik'::name,  
2 'porto_ndvi2'::name,'rast'::name);  
3
```

Data Output Messages Notifications



Eksport danych

Przykład 1 - ST_AsTiff

Query Query History

```
1 SELECT ST_AsTiff(ST_Union(rast))
2 FROM kowalik.porto_ndvi;
3
```

Data Output Messages Notifications

st_astiff
bytea

1 [binary data]

Przykład 2 - ST_AsGDALRaster

Query Query History

```
1 SELECT ST_AsGDALRaster(ST_Union(rast), 'GTiff', ARRAY['COMPRESS=DEFLATE',
2 'PREDICTOR=2', 'PZLEVEL=9'])
3 FROM kowalik.porto_ndvi;
4
```

Data Output Messages Notifications

st_asgdalraster
bytea

1 [binary data]

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

Query Query History

```
1 CREATE TABLE tmp_out AS
2 SELECT lo_from_bytea(0,
3 ST_AsGDALRaster(ST_Union(rast), 'GTiff', ARRAY['COMPRESS=DEFLATE',
4 'PREDICTOR=2', 'PZLEVEL=9']))
5 ) AS loid
6 FROM kowalik.porto_ndvi;
7 -----
8 SELECT lo_export(loid, 'C:myraster.tiff') --> Save the file in a place
9 -- where the user postgres have access. In windows a flash drive usually work
10 -- fine.
11 FROM tmp_out;
12 -----
13 SELECT lo_unlink(loid)
14 FROM tmp_out; --> Delete the large object.
15
```

Data Output Messages Notifications

lo_unlink
integer

1 1