



 cw_6/postgres@PostgreSQL 14



Query Query History



```
23 CREATE EXTENSION postgis;
24 CREATE EXTENSION postgis_raster;
25 ALTER SCHEMA schema_name RENAME TO Zmuda;
26
27 --Przykład 1 - ST_Intersects Przecięcie rastra z wektorem.
28 CREATE TABLE Zmuda.intersects AS
29 SELECT a.rast, b.municipality
30 FROM rasters.dem AS a, vectors.porto_parishes AS b
31 WHERE ST_Intersects(a.rast, b.geom) AND b.municipality ilike 'porto';
32
33 --Dodanie serial primary key:
34 alter table Zmuda.intersects add column rid SERIAL PRIMARY KEY;
```

Data output Messages Notifications



	rast raster	lock icon	municipality character varying (254)	lock icon
1	0100000...		PORTO	
2	0100000...		PORTO	
3	0100000...		PORTO	

Total rows: 25 of 25

Query complete 00:00:09.233

Ln 32, Col 1

cw_6/postgres@PostgreSQL 14



Query Query History

```
32
33 --Dodanie serial primary key:
34 alter table Zmuda.intersects add column rid SERIAL PRIMARY KEY;
35
36 --Utworzenie indeksu przestrzennego:
37 CREATE INDEX idx_intersects_rast_gist ON schema_name.intersects USING gist (ST_Conv
38
39 --Dodanie raster constraints:
40 -- schema::name table_name::name raster_column::name
41 SELECT AddRasterConstraints('Zmuda'::name,'intersects'::name,'rast'::name);
42
43 --Przykład 2 - ST_Clip
```

Data output Messages Notifications



	rast raster	municipality character varying (254)
--	----------------	---

1 0100000... PORTO

2 0100000... PORTO

3 0100000... PORTO

Total rows: 25 of 25

Query complete 00:00:09.233

Ln 33, Col 30



cw_6/postgres@PostgreSQL 14



No limit



Query Query History

```
42
43 --Przykład 2 - ST_Clip
44 --Obcinanie rastra na podstawie wektora.
45 CREATE TABLE Zmuda.clip AS
46 SELECT ST_Clip(a.rast, b.geom, true), b.municipality
47 FROM rasters.dem AS a, vectors.porto_parishes AS b
48 WHERE ST_Intersects(a.rast, b.geom) AND b.municipality like 'PORTO';
49
50 --Przykład 3 - ST_Union
51 --Połączenie wielu kafelków w jeden raster.
52 CREATE TABLE schema_name.union AS
53 SELECT ST_Union(ST_Clip(a.rast, b.geom, true))
```

Loading...

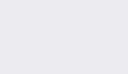
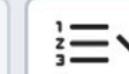
Data output Messages Notifications



Query returned successfully in 23 secs 409 msec.



cw_6/postgres@PostgreSQL 14



Query Query History



```
61 --Przykład pokazuje użycie funkcji ST_AsRaster w celu rastrowania tabeli z parafiami  
62 --o takiej samej charakterystyce przestrzennej  
63  
64 CREATE TABLE Zmuda.porto_parishes AS WITH r AS (  
65   SELECT rast FROM rasters.dem  
66   LIMIT 1 )  
67   SELECT ST_AsRaster(a.geom,r.rast,'8BUI',a.id,-32767) AS rast FROM vectors.porto_paris  
68 WHERE a.municipality ilike 'porto';  
69  
70 --Przykład 2 - ST_Union  
71 --Drugi przykład łączy rekordy z poprzedniego przykładu przy użyciu funkcji ST_UNION  
72 DROP TABLE schema_name.porto_parishes; --> drop table porto_parishes first CREATE T^P
```

Data output Messages Notifications



	rast raster	🔒
1	0100000...	
2	0100000...	
3	0100000...	

Total rows: 7 of 7

Query complete 00:00:07.817

Ln 64, Col 1



Query Query History



```
78
79 --Przykład 3 - ST_Tile
80 --Po uzyskaniu pojedynczego rastra można generować kafelki za pomocą funkcji ST_Tile.
81 DROP TABLE Zmuda.porto_parishes; --> drop table porto_parishes first
82 CREATE TABLE Zmuda.porto_parishes AS
83 WITH r AS (
84 SELECT rast FROM rasters.dem
85 LIMIT 1 )
86 SELECT st_tile(st_union(ST_AsRaster(a.geom,r.rast,'8BUI',a.id,- 32767)),128,128,true,
87 FROM vectors.porto_parishes AS a, r
88 WHERE a.municipality ilike 'porto';
89
90 --Konwertowanie rastrów na wektory (wektoryzowanie)
```

Data output Messages Notifications



Total rows: 8 of 8

Query complete 00:00:02.066

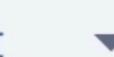
Ln 79, Col 20



cw_6/postgres@PostgreSQL 14



No limit



Query Query History

```
86 SELECT st_tile(st_union(ST_AsRaster(a.geom,r.rast,'8BUI',a.id,-32767)),128,128,true,
87 FROM vectors.porto_parishes AS a, r
88 WHERE a.municipality ilike 'porto';
89
90 --Konwertowanie rastrów na wektory (wektoryzowanie)
91 --Przykład 1 - ST_Intersection
92 create table Zmuda.intersection as
93 SELECT a.rid,(ST_Intersection(b.geom,a.rast)).geom,(ST_Intersection(b.geom,a.rast)) .
94 FROM rasters.landsat8 AS a, vectors.porto_parishes AS b
95 WHERE b.parish ilike 'paranhos' and ST_Intersects(b.geom,a.rast);
96
97 --Przykład 2 - ST_DumpAsPolygons
98 --ST_DumpAsPolygons konwertuje rastry w wektory (poligony).
```



Data output

Messages

Notifications



	rid [PK] integer	geom geometry	val double precision
1	221	0103000020B30...	10648
2	221	0103000020B30...	12155
3	221	0103000020B30...	9248

cw_6/postgres@PostgreSQL 14



Query Query History



```
97 --Przykład 2 - ST_DumpAsPolygons
98 --ST_DumpAsPolygons konwertuje rastry w wektory (poligony).
99
100 CREATE TABLE Zmuda.dumppolygons AS
101 SELECT
102 a.rid,(ST_DumpAsPolygons(ST_Clip(a.rast,b.geom))).geom,(ST_DumpAsPolygons(ST_Clip(a.r
103 FROM rasters.landsat8 AS a, vectors.porto_parishes AS b
104 WHERE b.parish ilike 'paranhos' and ST_Intersects(b.geom,a.rast));
105
106 --Analiza rastrów
107 --Przykład 1 - ST_Band
108 --Funkcja ST_Band służy do wyodrębniania pasm z rastra
```

Data output Messages Notifications



	rid [PK] integer	geom geometry
1	221	0103000020B30E000001000000050000009A6A639E2F5EE3C0E32C703B809504419A6A639E2F5EE3C0D19C81
2	221	0103000020B30E00000100000005000000D0EC61A7655AE3C0E32C703B80950441D0EC61A7655AE3C0D19C8

cw_6/postgres@PostgreSQL 14



Query Query History



```
103 FROM rasters.landsat8 AS a, vectors.porto_parishes AS b
104 WHERE b.parish ilike 'paranhos' and ST_Intersects(b.geom,a.rast);
105
106 --Analiza rastrów
107 --Przykład 1 - ST_Band
108 --Funkcja ST_Band służy do wyodrębniania pasm z rastra
109 CREATE TABLE Zmuda.landsat_nir AS
110 SELECT rid, ST_Band(rast,4) AS rast FROM rasters.landsat8;
111
112 --Przykład 2 - ST_Clip
113 CREATE TABLE schema_name.paranhos_dem AS
114 SELECT a.rid,ST_Clip(a.rast, b.geom,true) as rast
```

Data output Messages Notifications



	rid [PK] integer	rast raster	
1	1	0100000...	
2	2	0100000...	
3	3	0100000...	

cw_6/postgres@PostgreSQL 14



Query Query History



```
110 SELECT rid, ST_Band(rast,4) AS rast FROM rasters.landsat8;
111
112 --Przykład 2 - ST_Clip
113 CREATE TABLE Zmuda.paranhos_dem AS
114 SELECT a.rid,ST_Clip(a.rast, b.geom,true) as rast
115 FROM rasters.dem AS a, vectors.porto_parishes AS b
116 WHERE b.parish ilike 'paranhos' and ST_Intersects(b.geom,a.rast);
117
118 --Przykład 3 - ST_Slope
119 CREATE TABLE schema_name.paranhos_slope AS
120 SELECT a.rid,ST_Slope(a.rast,1,'32BF','PERCENTAGE') as rast FROM schema_name.paranhos
121
122 --Przykład 4 - ST_Reclass
```

Data output Messages Notifications

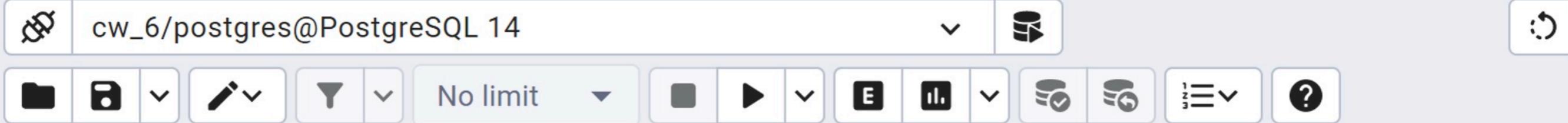


	rid [PK] integer	rast raster
1	380	0100001006172BF3E4D5A374080318D6907CA3EC09A49D3957D46E4C033B2707F2F9204410000000000000000
2	381	0100001006172BF3E4D5A374080318D6907CA3EC044951356C7ABE3C0DAE35DC0089604410000000000000000

Help ▾

Dashboard Properties SQL Statistics Dependencies Dependents [Zadanie_6.sql*](#)

 cw_6/postgres@PostgreSQL 14



Query Query History

2

```
116 WHERE b.parish ilike 'paranhos' and ST_Intersects(b.geom,a.rast);  
117  
118 --Przykład 3 - ST_Slope  
119 CREATE TABLE Zmuda.paranhos_slope AS  
120 SELECT a.rid,ST_Slope(a.rast,1,'32BF','PERCENTAGE') as rast  
121 FROM Zmuda.paranhos_dem AS a;  
122  
123 --Przykład 4 - ST_Reclass|  
124 --Aby zreklasyfikować raster należy użyć funkcji ST_Reclass.  
125 CREATE TABLE schema_name.paranhos_slope_reclass AS  
126 SELECT a.rid,ST_Reclass(a.rast,1,']0-15]:1, (15-30]:2, (30-9999:3',  
127 '32BF',0)  
128 FROM schema_name.paranhos_slope AS a.
```

>Loading...

Data output Messages Notifications

2



	rid	rast
1	380	0100001006172BF3E4D5A374080318D6907CA3EC09A49D3957D46E4C033B2707F2F9204410000000000000000
2	381	0100001006172BF3E4D5A374080318D6907CA3EC044951356C7ABE3C0DAE35DC0089604410000000000000000

Total rows: 4 of 4

Query complete 00:00:08.760

Ln 123, Col 26

pgAdmin File Object Tools Help

Browser FTS Templates Foreign Tables Functions Materialized Views Operators Procedures Sequences Tables (13)

cw_6/postgres@PostgreSQL 14

No limit

Query History

```
120 SELECT a.rid,ST_Slope(a.rast,1,'32BF','PERCENTAGE') as rast
121 FROM Zmuda.paranhos_dem AS a;
122
123 --Przykład 4 - ST_Reclass
124 --Aby zreklasyfikować raster należy użyć funkcji ST_Reclass.
125 CREATE TABLE Zmuda.paranhos_slope_reclass AS
126 SELECT a.rid,ST_Reclass(a.rast,1,']0-15]:1, (15-30]:2, (30-9999:3',
127 '32BF',0)
128 FROM Zmuda.paranhos_slope AS a;
129
130 --Przykład 5 - ST_SummaryStats
131 --Aby obliczyć statystyki rastra można użyć funkcji ST_SummaryStats.
132 SELECT * FROM Zmuda.paranhos_slope_reclass AS a;
```

Data output Messages Notifications

	rid integer	st_reclass raster
1	380	01000001...
2	381	01000001...
3	382	01000001...

Total rows: 4 of 4 Query complete 00:00:02.875

Ln 125, Col 45

pgAdmin File Object Tools Help

Browser FTS Templates Foreign Tables Functions Materialized Views Operators Procedures Sequences Tables (13)

Dashboard Properties SQL Statistics Dependencies Dependents Zadanie_6.sql*

Query History

```
125 CREATE TABLE zmuda.parannos_slope_reclass AS
126 SELECT a.rid,ST_Reclass(a.rast,1,']0-15]:1, (15-30]:2, (30-9999:3',
127 '32BF',0)
128 FROM Zmuda.paranhos_slope AS a;
129
130 --Przykład 5 - ST_SummaryStats
131 --Aby obliczyć statystyki rastra można użyć funkcji ST_SummaryStats.
132 SELECT st_summarystats(a.rast) AS stats FROM Zmuda.paranhos_dem AS a;
133
134 --Przykład 6 - ST_SummaryStats oraz Union
135 --Przy użyciu UNION można wygenerować jedną statystykę wybranego rastra.
136 SELECT st_summarystats(ST_Union(a.rast)) FROM schema_name.paranhos_dem AS a;
137
```

>Loading...

Data output Messages Notifications

	stats
1	summarystats
1	(2616,278385,...
2	(6463,816615,...
3	(682,95581,14...

Total rows: 4 of 4 Query complete 00:00:02.017

Ln 133, Col 1

Help ▾

Dashboard Properties SQL Statistics Dependencies Dependents **Zadanie_6.sql***

cw_6/postgres@PostgreSQL 14

No limit

Query History

```
127 '32BF',0)
128 FROM Zmuda.paranhos_slope AS a;

129
130 --Przykład 5 - ST_SummaryStats
131 --Aby obliczyć statystyki rastra można użyć funkcji ST_SummaryStats.
132 SELECT st_summarystats(a.rast) AS stats FROM Zmuda.paranhos_dem AS a;

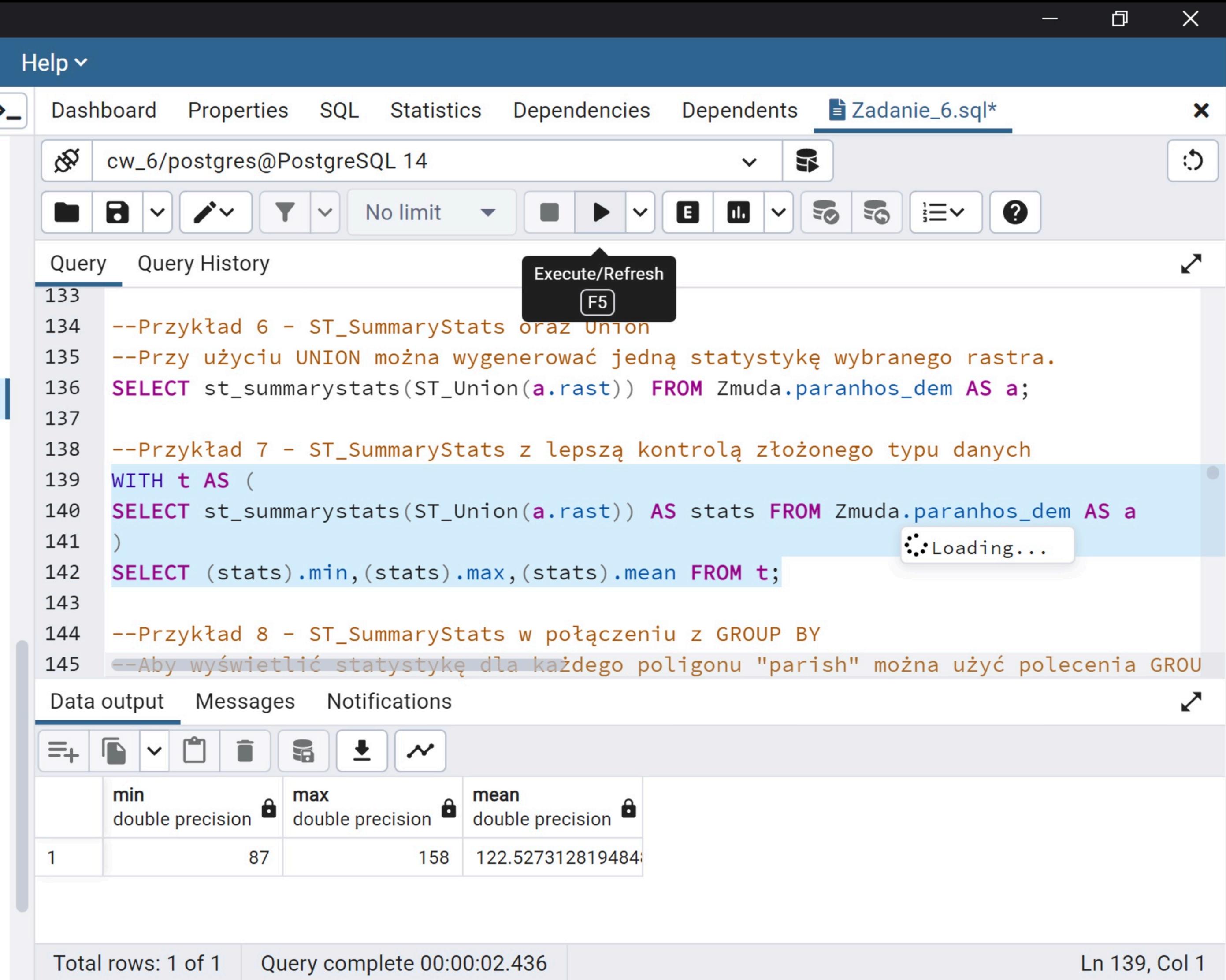
133
134 --Przykład 6 - ST_SummaryStats oraz Union
135 --Przy użyciu UNION można wygenerować jedną statystykę wybranego rastra.
136 SELECT st_summarystats(ST_Union(a.rast)) FROM Zmuda.paranhos_dem AS a;

137
138 --Przykład 7 - ST_SummaryStats z lepszą kontrolą złożonego typu danych
139 WITH t AS (
```

Data output Messages Notifications

st_summarystats 

	st_summarystats
summarystats	
1	(9977,1222455,1...



pgAdmin File Object Tools Help

Browser FTS Templates Foreign Tables Functions Materialized Views Operators Procedures Sequences Tables (13)

Dashboard Properties SQL Statistics Dependencies Dependents Zadanie_6.sql*

Query History

```
142 SELECT (stats).min,(stats).max,(stats).mean FROM t,
143
144 --Przykład 8 - ST_SummaryStats w połączeniu z GROUP BY
145 --Aby wyświetlić statystykę dla każdego poligona "parish" można użyć polecenia GROUP
146 WITH t AS (
147     SELECT b.parish AS parish, st_summarystats(ST_Union(ST_Clip(a.rast,
148         b.geom,true))) AS stats
149     FROM rasters.dem AS a, vectors.porto_parishes AS b
150     WHERE b.municipality ilike 'porto' and ST_Intersects(b.geom,a.rast) group by b.parish
151 )
152     SELECT parish,(stats).min,(stats).max,(stats).mean FROM t;
153
154 --Przykład 9 - ST_Value
```

Data output Messages Notifications

	parish	min	max	mean
	character varying (254)	double precision	double precision	double precision
1	Bonfim	1	159	107.5658842667906
2	Campanhã	0	178	74.66732213085449
3	Paranhos	87	158	122.52731281948482

Total rows: 7 of 7 Query complete 00:00:02.373 Ln 146, Col 1

pgAdmin

File ▾ Object ▾ Tools ▾ Help ▾

Browser



- > FTS Templates
- > Foreign Tables
- > Functions
- > Materialized Views
- > Operators
- > Procedures
- > Sequences
- > Tables (13)

Dashboard Properties SQL Statistics Dependencies Dependents Zadanie_6.sql*

cw_6/postgres@PostgreSQL 14

Query Query History

```
149 FROM rasters.dem AS a, vectors.porto_parishes AS b
150 WHERE b.municipality ilike 'porto' and ST_Intersects(b.geom,a.rast) group by b.parish
151 )
152 SELECT parish,(stats).min,(stats).max,(stats).mean FROM t;
153
154 --Przykład 9 - ST_Value
155 --Funkcja ST_Value pozwala wyodrębnić wartość piksela z punktu lub zestawu punktów.
156 --Poniższy przykład wyodrębnia punkty znajdujące się w tabeli vectors.places.
157 SELECT b.name,st_value(a.rast,(ST_Dump(b.geom)).geom) FROM
158 rasters.dem a, vectors.places AS b
159 WHERE ST_Intersects(a.rast,b.geom)
160 ORDER BY b.name;
```

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

Total rows: 33 of 33 Query complete 00:00:06.394

Ln 157, Col 1

pgAdmin File Object Tools Help

Browser FTS Templates Foreign Tables Functions Materialized Views Operators Procedures Sequences Tables (13)

Dashboard Properties SQL Statistics Dependencies Dependents Zadanie_6.sql*

Query History

```
167 SELECT AddRasterConstraints('Zmuda'::name, 'tpi30'::name, 'rast'::name);
168
169 --problem do samodzielnego rozwiazania
170 CREATE TABLE Zmuda.tpi30_porto AS
171 SELECT ST_TPI(a.rast,1) AS rast
172 FROM rasters.dem AS a, vectors.porto_parishes AS b
173 WHERE ST_Intersects(a.rast, b.geom) AND b.municipality ILIKE 'porto';
174
175 CREATE INDEX idx_tpi30_porto_rast_gist ON Zmuda.tpi30_porto USING gist (ST_ConvexHull
176
177 SELECT AddRasterConstraints('Zmuda'::name, 'tpi30_porto'::name, 'rast'::name);
178
179 --Algebra map
```

Data output Messages Notifications

	rast raster	lock
1	0100000...	
2	0100000...	
3	0100000...	

Total rows: 25 of 25 Query complete 00:00:28.114 Ln 175, Col 35

pgAdmin File Object Tools Help

Browser FTS Templates Foreign Tables Functions Materialized Views Operators Procedures Sequences Tables (13)

FTS Templates Foreign Tables Functions Materialized Views Operators Procedures Sequences Tables (13)

Dashboard Properties SQL Statistics Dependencies Dependents Zadanie_6.sql*

cw_6/postgres@PostgreSQL 14

No limit

Query History

177 **SELECT** AddRasterConstraints('Zmuda'::name,'tpi30_porto'::name,'rast'::name);
178
179 --Algebra map
180 **CREATE TABLE** Zmuda.porto_ndvi **AS**
181 **WITH r AS** (**SELECT** a.rid,ST_Clip(a.rast, b.geom,true) **AS** rast
182 **FROM** rasters.landsat8 **AS** a, vectors.porto_parishes **AS** b
183 **WHERE** b.municipality **ilike** 'porto' **and** ST_Intersects(b.geom,a.rast))
184 **SELECT** r.rid,ST_MapAlgebra(r.rast, 1,r.rast, 4,
185 '([rast2.val] - [rast1.val]) / ([rast2.val] + [rast1.val]):float','32BF') **AS** rast
186 **FROM** r;
187
188 **CREATE INDEX** idx_porto_ndvi_rast_gist **ON** osika.porto_ndvi **USING** gist (ST_ConvexHull(rast))

Data output Messages Notifications

	rid [PK] integer	rast raster
1	245	0100000...
2	246	0100000...
3	270	0100000...

Total rows: 23 of 23 Query complete 00:00:02.184

Ln 180, Col 32

Browser



- > FTS Templates
- > Foreign Tables
- > Functions
- > Materialized Views
- > Operators
- > Procedures
- > Sequences
- Tables (13)

- > clip
- > dumppolygons
- > intersection
- > intersects
- > landsat_nir
- > paranhos_dem

Dashboard Properties SQL Statistics Dependencies Dependents Zadanie_6.sql*

cw_6/postgres@PostgreSQL 14



Query Query History

```
180 CREATE TABLE Zmuda.porto_ndvi AS
181 WITH r AS (SELECT a.rid,ST_Clip(a.rast, b.geom,true) AS rast
182 FROM rasters.landsat8 AS a, vectors.porto_parishes AS b
183 WHERE b.municipality ilike 'porto' and ST_Intersects(b.geom,a.rast))
184 SELECT r.rid,ST_MapAlgebra(r.rast, 1,r.rast, 4,
185 '([rast2.val] - [rast1.val]) / ([rast2.val] + [rast1.val])::float','32BF') AS rast
186 FROM r;
187
188 CREATE INDEX idx_porto_ndvi_rast_gist ON Zmuda.porto_ndvi USING gist (ST_ConvexHull(rast));
189
190 SELECT AddRasterConstraints('Zmuda'::name,'porto_ndvi'::name,'rast'::name);
```

pgAdmin File Object Tools Help

Browser FTS Templates Foreign Tables Functions Materialized Views Operators Procedures Sequences Tables (13)

Dashboard Properties SQL Statistics Dependencies Dependents Zadanie_6.sql*

Query History

```
194 create or replace function Zmuda.ndvi(
195     value double precision [] [] [], pos integer [][],
196     VARIADIC userargs text []
197 )
198 RETURNS double precision AS $$
```

>Loading...

```
199 BEGIN
200 --RAISE NOTICE 'Pixel Value: %', value [1][1][1];-->For debug purposes
201 RETURN (value [2][1][1] - value [1][1][1])/ (value [2][1][1]+value [1][1][1]); --> NDVI
202 END;
203 $$
```

204 LANGUAGE 'plpgsql' IMMUTABLE COST 1000;

205

Data output Messages Notifications

CREATE FUNCTION

Query returned successfully in 2 secs 191 msec.

Total rows: 23 of 23 Query complete 00:00:02.191 Ln 194, Col 1

Browser



Dashboard Properties SQL Statistics Dependencies Dependents [Zadanie_6.sql*](#)

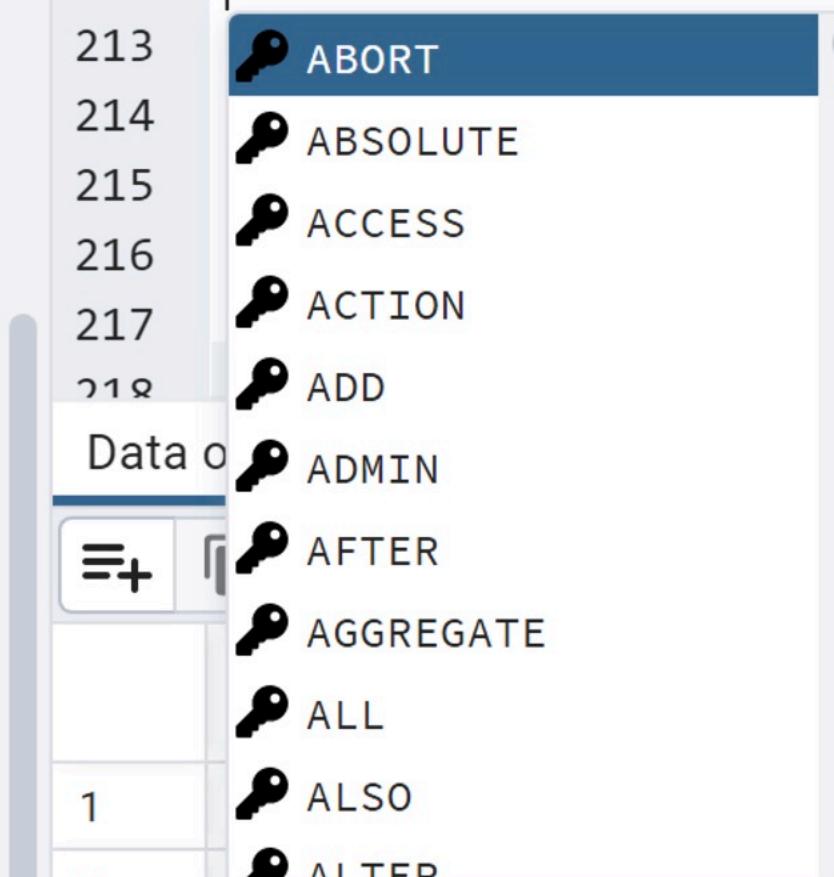
- >  FTS Templates
 - >  Foreign Tables
 - >  Functions
 - >  Materialized Views
 - >  Operators
 - >  Procedures
 - >  1..3 Sequences
 - >  Tables (13)
 - >  clip
 - >  dumppolygons
 - >  intersection
 - >  intersects
 - >  landsat_nir
 - >  paranhos_dem
 - >  paranhos_slope
 - >  paranhos_slope_recl
 - >  porto_ndvi
 - >  porto_ndvi2
 - >  tpi30
 - >  tpi30_porto
 - >  union
 - >  Trigger Functions
 - >  Types

 cw_6/postgres@PostgreSQL 14  



Query Query History

```
206 --W kwerendzie algebra map należy można wywołać zdefiniowaną wcześniej funkcję:  
207 CREATE TABLE Zmuda.porto_ndvi2 AS  
208 WITH r AS (SELECT a.rid,ST_Clip(a.rast, b.geom,true) AS rast FROM rasters.landsat8 AS a  
209 WHERE b.municipality ilike 'porto' AND ST_Intersects(b.geom,a.rast))  
210 SELECT r.rid,ST_MapAlgebra(r.rast, ARRAY[1,4], 'Zmuda.ndvi(double precision[],integer[]))::text) AS rast FROM r;  
211  
212 |
```



Total rows: 23 of 23 Query complete 00:00:16.419

Ln 212, Col 1

pgAdmin File Object Tools Help

Browser FTS Templates Foreign Tables Functions Materialized Views Operators Procedures Sequences Tables (13)

Dashboard Properties SQL Statistics Dependencies Dependents Zadanie_6.sql*

Query History

```
--W kwerendzie algebra map należy można wywołać zdefiniowaną wcześniej funkcję:  
CREATE TABLE Zmuda.porto_ndvi2 AS  
WITH r AS (SELECT a.rid,ST_Clip(a.rast, b.geom,true) AS rast FROM rasters.landsat8 AS a  
WHERE b.municipality ilike 'porto' AND ST_Intersects(b.geom,a.rast))  
SELECT r.rid,ST_MapAlgebra(r.rast, ARRAY[1,4], 'Zmuda.ndvi(double precision[],integer  
'32BF)::text) AS rast FROM r;  
  
CREATE INDEX idx_porto_ndvi2_rast_gist ON Zmuda.porto_ndvi2 USING gist (ST_ConvexHull);  
  
SELECT AddRasterConstraints('Zmuda'::name,'porto_ndvi2'::name,'rast'::name);  
  
--Eksport danych  
--Przykład 1 - ST_AsTiff
```

Data output Messages Notifications

	rid [PK] integer	rast raster
1	245	0100000...
2	246	0100000...
3	270	0100000...

Total rows: 23 of 23 Query complete 00:00:16.419 Ln 217, Col 17

pgAdmin File Object Tools Help

Browser FTS Templates Foreign Tables Functions Materialized Views Operators Procedures Sequences Tables (13) clip dumppolygons intersection intersects landsat_nir paranhos_dem paranhos_slope paranhos_slope_reclass porto_ndvi porto_ndvi2 tpi30 tpi30_porto union Trigger Functions Types

Dashboard Properties SQL Statistics Dependencies Dependents Zadanie_6.sql*

cw_6/postgres@PostgreSQL 14

No limit

Query History

215 `SELECT AddRasterConstraints('Zmuda'::name, 'porto_ndvi2'::name, 'rast'::name);`
216
217 --Eksport danych
218 --Przykład 1 - ST_AsTiff
219 `SELECT ST_AsTiff(ST_Union(rast)) FROM Zmuda.porto_ndvi;`
220
221 --Przykład 2 - ST_AsGDALRaster
222 `SELECT ST_AsGDALRaster(ST_Union(rast), 'GTiff', ARRAY['COMPRESS=DEFLATE', 'PREDICTOR=2']);`
223 `FROM Zmuda.porto_ndvi;`
224 Loading...
225 `SELECT ST_GDALDrivers();`
226
227 --Przykład 3 - Zapisywanie danych na dysku za pomocą dużego obiektu

Data output Messages Notifications

	st_astiff	bytea
1	[binary da...]	lock

Total rows: 1 of 1 Query complete 00:00:05.951

Ln 219, Col 1

pgAdmin File Object Tools Help

Browser

- > FTS Templates
- > Foreign Tables
- > Functions
- > Materialized Views
- > Operators
- > Procedures
- > Sequences
- > Tables (13)

- > clip
- > dumppolygons
- > intersection
- > intersects
- > landsat_nir
- > paranhos_dem
- > paranhos_slope
- > paranhos_slope_reclass
- > porto_ndvi
- > porto_ndvi2
- > tpi30
- > tpi30_porto
- > union

- > Trigger Functions
- > Types

Dashboard Properties SQL Statistics Dependencies Dependents Zadanie_6.sql*

cw_6/postgres@PostgreSQL 14

No limit ▾

Query History

```
215 SELECT AddRasterConstraints('Zmuda'::name,'porto_ndvi2'::name,'rast'::name);
216
217 --Eksport danych
218 --Przykład 1 - ST_AsTiff
219 SELECT ST_AsTiff(ST_Union(rast)) FROM Zmuda.porto_ndvi;
220
221 --Przykład 2 - ST_AsGDALRaster
222 SELECT ST_AsGDALRaster(ST_Union(rast), 'GTiff', ARRAY['COMPRESS=DEFLATE', 'PREDICTOR=3']);
223 FROM Zmuda.porto_ndvi;
224
225 SELECT ST_GDALDrivers();
226
227 --Przykład 3 - Zapisywanie danych na dysku za pomocą dużego obiektu
```

Data output Messages Notifications

	st_asgdalraster	bytea
1	[binary data]	lock

Total rows: 1 of 1 Query complete 00:00:14.859

Ln 222, Col 1

pgAdmin File Object Tools Help

Browser FTS Templates Foreign Tables Functions Materialized Views Operators Procedures Sequences Tables (13)

clip dumppolygons intersection intersects landsat_nir paranhos_dem paranhos_slope paranhos_slope_reclass porto_ndvi porto_ndvi2 tpi30 tpi30_porto union Trigger Functions Types

Dashboard Properties SQL Statistics Dependencies Dependents Zadanie_6.sql*

cw_6/postgres@PostgreSQL 14

No limit

Query History

```
215 SELECT AddRasterConstraints('Zmuda'::name,'porto_ndvi2'::name,'rast'::name);
216
217 --Eksport danych
218 --Przykład 1 - ST_AsTiff
219 SELECT ST_AsTiff(ST_Union(rast)) FROM Zmuda.porto_ndvi;
220
221 --Przykład 2 - ST_AsGDALRaster
222 SELECT ST_AsGDALRaster(ST_Union(rast), 'GTiff', ARRAY['COMPRESS=DEFLATE', 'PREDICTOR=2']);
223 FROM Zmuda.porto_ndvi;
224
225 SELECT ST_GDALDrivers();
226
227 --Przykład 3 - Zapisywanie danych na dysku za pomocą dużego obiektu
```

Data output Messages Notifications

	st_gdaldrivers record	lock
1	(0,GTiff,GeoTIFF)	
2	(1,AAGrid,"Arbitrary Grid")	
3	(2,DTED,"DTED")	

Total rows: 8 of 8 Query complete 00:00:07.493 Ln 225, Col 1

pgAdmin File Object Tools Help

Browser

- > FTS Templates
- > Foreign Tables
- > Functions
- > Materialized Views
- > Operators
- > Procedures
- > Sequences
- > Tables (13)
 - > clip
 - > dumppolygons
 - > intersection
 - > intersects
 - > landsat_nir
 - > paranhos_dem
 - > paranhos_slope
 - > paranhos_slope_reclass
 - > porto_ndvi
 - > porto_ndvi2
 - > tpi30
 - > tpi30_porto
 - > union
- > Trigger Functions

Dashboard Properties SQL Statistics Dependencies Dependents Zadanie_6.sql*

cw_6/postgres@PostgreSQL 14

No limit

Query History

--Przykład 3 - Zapisywanie danych na dysku za pomocą dużego obiektu

```
227 CREATE TABLE tmp_out AS
228 SELECT lo_from_byter(0,
229 ST_AsGDALRaster(ST_Union(rast), 'GTiff', ARRAY['COMPRESS=DEFLATE',
230 'PREDICTOR=2', 'PZLEVEL=9']) ) AS loid
231 FROM Zmuda.porto_ndvi;
232 -----
233 SELECT lo_export(loid, '\\Mac\\Home\\Desktop\\Dane - Ćwiczenia 6\\rasters\\myraster.tif')
234 FROM tmp_out;
235 -----
236 SELECT lo_unlink(loid)
237 FROM tmp_out; --> Delete the large object.
```

Data output Messages Notifications

	loid	oid	lock
1		48349	