

Przykład 1 – ST_Intersects

```
1 CREATE EXTENSION postgis;
2 CREATE EXTENSION postgis_raster;
3
4 CREATE TABLE rasters.dem (
5     rid SERIAL PRIMARY KEY,
6     rast RASTER
7 );
8
9 -- C:\Program Files\PostgreSQL\17\bin>psql -d BDP6 -U postgres
10 -- -h localhost -p 5432 -f "C:\\Users\\[redacted]
11 -- bazy danych przestrzennych\\cwiczenia\\bdp_cw6\\PostGIS raster - dane\\dem.sql"
12
13 SELECT * FROM rasters.dem;
14
15 CREATE TABLE schema_name.intersects AS
16 SELECT a.rast, b.municipality
17 FROM rasters.dem AS a, vectors.porto_parishes AS b
18 WHERE ST_Intersects(a.rast, b.geom) AND b.municipality ilike 'porto';
19
20 SELECT
21     ST_Width(rast) AS width,
22     ST_Height(rast) AS height,
23     ST_SRID(rast) AS srid,
24     ST_NumBands(rast) AS num_bands
25 FROM schema_name.intersects;
```

Data Output Messages Notifications

	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

Total rows: 25 of 25 Query complete 00:00:00.158 Ln 25, Col 4

```

29 -- 1. dodanie serial primary key:
30 ALTER TABLE schema_name.intersects
31 ADD COLUMN rid SERIAL PRIMARY KEY;
32
33 -- 2. utworzenie indeksu przestrzennego:
34 CREATE INDEX idx_intersects_rast_gist ON schema_name.intersects
35 USING gist (ST_ConvexHull(rast));
36
37 -- 3. dodanie raster constraints: -- schema::name table_name::name raster_column::name
38 SELECT AddRasterConstraints('schema_name'::name,
39 'intersects'::name, 'rast'::name);
40
41 -- Sprawdzenie, czy ograniczenia rastrowe zostały dodane
42 SELECT r_table_schema, r_table_name, r_raster_column
43 FROM raster_columns
44 WHERE r_table_schema = 'schema_name' AND r_table_name = 'intersects' AND r_raster_column = 'rast';
45

```

Data Output Messages Notifications

	r_table_schema name	r_table_name name	r_raster_column name
1	schema_name	intersects	rast

Przykład 2 – ST_Clip

```

47 CREATE TABLE schema_name.clip AS
48 SELECT ST_Clip(a.rast, b.geom, true) AS rast, b.municipality
49 FROM rasters.dem AS a, vectors.porto_parishes AS b
50 WHERE ST_Intersects(a.rast, b.geom) AND b.municipality like 'PORTO';
51
52 SELECT
53     ST_Width(rast) AS width,
54     ST_Height(rast) AS height,
55     ST_SRID(rast) AS srid,
56     ST_NumBands(rast) AS num_bands
57 FROM schema_name.clip;
58

```

Data Output Messages Notifications

	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
6	31	83	3763	1
7	15	32	3763	1
8	66	73	3763	1
9	35	15	3763	1

Total rows: 25 of 25

Query complete 00:00:00.135

Ln 54, Col 19

Przykład 3 – ST_Union

```
60 -- Przykład 3 - ST_Union
61 CREATE TABLE schema_name.union AS
62 SELECT ST_Union(ST_Clip(a.rast, b.geom, true))
63 FROM rasters.dem AS a, vectors.porto_parishes AS b
64 WHERE b.municipality ilike 'porto' and ST_Intersects(b.geom,a.rast);
65
66 SELECT
67     ST_Width(st_union) AS width,
68     ST_Height(st_union) AS height,
69     ST_SRID(st_union) AS srid,
70     ST_NumBands(st_union) AS num_bands
71 FROM schema_name.union;
72
```

Data Output Messages Notifications

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

Tworzenie rastrów z wektorów (rastrowanie)

Przykład 1 - ST_AsRaster

```
73 -- Przykład 1 - ST_AsRaster
74 v CREATE TABLE schema_name.porto_parishes AS
75 WITH r AS (
76 SELECT rast FROM rasters.dem
77 LIMIT 1
78 )
79 SELECT ST_AsRaster(a.geom,r.rast,'8BUI',a.id,-32767) AS rast
80 FROM vectors.porto_parishes AS a, r
81 WHERE a.municipality ilike 'porto';
82
83 v SELECT
84     ST_Width(rast) AS width,
85     ST_Height(rast) AS height,
86     ST_SRID(rast) AS srid,
87     ST_NumBands(rast) AS num_bands
88 FROM schema_name.porto_parishes;|
89
```

Data Output Messages Notifications

SQL

	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

Przykład 2 - ST_Union

```
90 -- Przykład 2 - ST_Union
91 DROP TABLE schema_name.porto_parishes; --> drop table porto_parishes first
92 CREATE TABLE schema_name.porto_parishes AS
93 WITH r AS (
94 SELECT rast FROM rasters.dem
95 LIMIT 1
96 )
97 SELECT st_union(ST_AsRaster(a.geom,r.rast,'8BUI',a.id,-32767)) AS rast
98 FROM vectors.porto_parishes AS a, r
99 WHERE a.municipality ilike 'porto';
100 SELECT
101     ST_Width(rast) AS width,
102     ST_Height(rast) AS height,
103     ST_SRID(rast) AS srid,
104     ST_NumBands(rast) AS num_bands
105 FROM schema_name.porto_parishes;
```

Data Output Messages Notifications

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

Przykład 3 - ST_Tile

```
107 -- Przykład 3 - ST_Tile
108 DROP TABLE schema_name.porto_parishes; --> drop table porto_parishes first
109 CREATE TABLE schema_name.porto_parishes AS
110 WITH r AS (
111 SELECT rast FROM rasters.dem
112 LIMIT 1 )
113 SELECT st_tile(st_union(ST_AsRaster(a.geom,r.rast,'8BUI',a.id,
114 32767)),128,128,true,-32767) AS rast
115 FROM vectors.porto_parishes AS a, r
116 WHERE a.municipality ilike 'porto';
117
118 SELECT
119     ST_Width(rast) AS width,
120     ST_Height(rast) AS height,
121     ST_SRID(rast) AS srid,
122     ST_NumBands(rast) AS num_bands
123 FROM schema_name.porto_parishes;
```

Data Output Messages Notifications

	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

Total rows: 8 of 8 Query complete 00:00:00.103 Ln 122, Col 26

Data Output	Messages	Notifications
	rid integer	st_astext text
1	366	POLYGON((-40817.32136349093 168178.4421137896,-40817.32136349093 168148.73631987217,-40787.009961412616 168148.73631987217,-40787.009961412616 168178.4421137896,-40817.32136349093 168178.4421137896))
2	366	POLYGON((-40847.63276556926 168148.73631987217,-40847.63276556926 168119.03052595473,-40817.32136349093 168119.03052595473,-40817.32136349093 168148.73631987217,-40847.63276556926 168148.73631987217))
3	366	POLYGON((-40817.32136349093 168148.73631987217,-40817.32136349093 168119.03052595473,-40787.009961412616 168119.03052595473,-40787.009961412616 168148.73631987217,-40817.32136349093 168148.73631987217))
4	366	POLYGON((-40877.944167647576 168119.03052595473,-40877.944167647576 168089.3247320373,-40847.63276556926 168089.3247320373,-40847.63276556926 168119.03052595473,-40877.944167647576 168119.03052595473))
5	366	POLYGON((-40847.63276556926 168119.03052595473,-40847.63276556926 168089.3247320373,-40817.32136349093 168089.3247320373,-40817.32136349093 168119.03052595473,-40847.63276556926 168119.03052595473))
6	366	POLYGON((-40817.32136349093 168119.03052595473,-40817.32136349093 168089.3247320373,-40787.009961412616 168089.3247320373,-40787.009961412616 168119.03052595473,-40817.32136349093 168119.03052595473))
Total rows: 1000 of 6442 Query complete 00:00:00.145 Ln 151, Col 23		

Analiza rastrów

Przykład 1 - ST_Band

```
154 -- Przykład 1 - ST_Band
155 v CREATE TABLE schema_name.landsat_nir AS
156 SELECT rid, ST_Band(rast,4) AS rast
157 FROM rasters.landsat8;
158
159 v SELECT
160     ST_Width(rast) AS width,
161     ST_Height(rast) AS height,
162     ST_SRID(rast) AS srid,
163     ST_NumBands(rast) AS num_bands
164 FROM schema_name.landsat_nir;
```

Data Output Messages Notifications

	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

Total rows: 630 of 630 Query complete 00:00:00.116 Ln 163, Col 9

Przykład 2 - ST_Clip

```

165 -- Przykład 2 - ST_Clip
166 CREATE TABLE schema_name.paranhos_dem AS
167 SELECT a.rid,ST_Clip(a.rast, b.geom,true) AS rast
168 FROM rasters.dem AS a, vectors.porto_parishes AS b
169 WHERE b.parish ilike 'paranhos' and ST_Intersects(b.geom,a.rast);
170
171 SELECT
172     ST_Width(rast) AS width,
173     ST_Height(rast) AS height,
174     ST_NumBands(rast) AS num_bands
175 FROM schema_name.paranhos_dem;

```

[Data Output](#) [Messages](#) [Notifications](#)

Navigation icons: Home, Back, Forward, Search, etc.

	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

```
177 -- Przykład 3 - ST_Slope
178 v CREATE TABLE schema_name.paranhos_slope AS
179 SELECT a.rid,ST_Slope(a.rast,1,'32BF','PERCENTAGE') as rast
180 FROM schema_name.paranhos_dem AS a;
181
182 SELECT * FROM schema_name.paranhos_slope;
183
```

Data Output Messages Notifications

[illegible]

Przykład 4 - ST_Reclass

```
184 -- Przykład 4 - ST_Reclass
185 CREATE TABLE schema_name.paranhos_slope_reclass AS
186 SELECT a.rid, ST_Reclass(a.rast, 1, ']0-15]:1, (15-30]:2, (30-9999:3', '32BF', 0)
187 FROM schema_name.paranhos_slope AS a;
188
189 SELECT * FROM schema_name.paranhos_slope_reclass;
```

[Data Output](#) [Messages](#) [Notifications](#)

	rid integer	st_reclass raster
1	380	010000001006172BF3E4D5A374080318D6907CA3EC09A49D3957D46E4C033B2707F2F92044100000000000000
2	382	010000001006172BF3E4D5A374080318D6907CA3EC02E3C8390DE87E2C0D7D06D6CAD85044100000000000000
3	412	010000001006172BF3E4D5A374080318D6907CA3EC0187635E2BF88E3C0474F11FE054A044100000000000000
4	381	010000001006172BF3E4D5A374080318D6907CA3EC044951356C7ABE3C0DAE35DC00896044100000000000000

Przykład 5 - ST SummaryStats

```
191 -- Przykład 5 - ST_SummaryStats
192 ✓ SELECT st_summarystats(a.rast) AS stats
193 FROM schema_name.paranhos_dem AS a;
```

[Data Output](#) [Messages](#) [Notifications](#)

	stats summarystats
1	(2616,278385,106.41628440366972,11.622628762211638,87,14...
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

```
195 -- Przykład 6 - ST_SummaryStats oraz Union
196 ✓ SELECT st_summarystats(ST_Union(a.rast))
197 FROM schema_name.paranhos_dem AS a;
```

[Data Output](#) [Messages](#) [Notifications](#)

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

Przykład 7 - ST_SummaryStats z lepszą kontrolą złożonego typu danych

```
199 -- Przykład 7 - ST_SummaryStats z lepszą kontrolą złożonego typu danych
200 WITH t AS (
201     SELECT st_summarystats(ST_Union(a.rast)) AS stats
202     FROM schema_name.paranhos_dem AS a
203 )
204 SELECT (stats).min,(stats).max,(stats).mean FROM t;
```

Data Output Messages Notifications

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

Przykład 8 - ST_SummaryStats w połączeniu z GROUP BY

```
206 -- Przykład 8 - ST_SummaryStats w połączeniu z GROUP BY
207 WITH t AS (
208     SELECT b.parish AS parish, st_summarystats(ST_Union(ST_Clip(a.rast,
209     b.geom,true))) AS stats
210     FROM rasters.dem AS a, vectors.porto_parishes AS b
211     WHERE b.municipality ilike 'porto' and ST_Intersects(b.geom,a.rast)
212     group by b.parish
213 )
214 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
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

Przykład 9 - ST_Value

```
216 -- Przykład 9 - ST_Value
217 v SELECT b.name, st_value(a.rast,(ST_Dump(b.geom)).geom)
218 FROM rasters.dem a, vectors.places AS b
219 WHERE ST_Intersects(a.rast,b.geom)
220 ORDER BY b.name;
221
222
```

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: 33 of 33 Query complete 00:00:00.094 Ln 221, Col 1

Przykład 10 - ST_TPI

```
222 -- Przykład 10 - ST_TPI
223 v CREATE TABLE schema_name.tpi30 as
224 SELECT ST_TPI(a.rast, 1) AS rast
225 FROM rasters.dem a;
226
227 v SELECT st_summarystats(tpi.rast)
228 FROM schema_name.tpi30 AS tpi;
```

Data Output Messages Notifications



	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)

Total rows: 589 of 589 Query complete 00:00:00.371 Ln 227, Col 27

```
230 -- Poniższa kwerenda utworzy indeks przestrzenny:
231 CREATE INDEX idx_tpi30_rast_gist ON schema_name.tpi30
232 USING gist (ST_ConvexHull(rast));
233
```

Data Output Messages Notifications

CREATE INDEX

Query returned successfully in 81 msec.

```
234 -- Dodanie constraintów:
235 SELECT AddRasterConstraints('schema_name'::name,
236 'tpi30'::name, 'rast'::name);
```

Data Output Messages Notifications

<div><div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div>SQL</div></div></div>	
	<div><div>addrasterconstraints</div><div>boolean</div><div></div></div>
1	<div><div>true</div></div>

Obliczanie TPI na mniejszym obszarze

```
193 -- Optymalizacja
194 ▾ CREATE TABLE schema_name.tpi30_porto AS
195 SELECT ST_TPI(a.rast,1) AS rast
196 FROM rasters.dem AS a, vectors.porto_parishes AS b
197 WHERE ST_Intersects(a.rast, b.geom) AND b.municipality ilike 'porto';
198
199 ▾ CREATE INDEX idx_tpi30_porto_rast_gist ON schema_name.tpi30_porto
200 USING gist (ST_ConvexHull(rast));
201
202 ▾ SELECT AddRasterConstraints('schema_name'::name,
203 'tpi30_porto'::name, 'rast'::name);
204
205 ▾ SELECT st_summarystats(a.rast)
206 FROM schema_name.tpi30_porto AS a;
207
208 -- Eksport danych
```

Data Output Messages Notifications



	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)	
9	(10000,0,0,1.9106535203955748,-14.5,20.625)	

Total rows: 25 of 25 Query complete 00:00:00.160 Ln 205, Col 25

```
199 ▾ CREATE INDEX idx_tpi30_porto_rast_gist ON schema_name.tpi30_porto
200 USING gist (ST_ConvexHull(rast));
201
202 ▾ SELECT AddRasterConstraints('schema_name'::name,
203 'tpi30_porto'::name, 'rast'::name);
```

Data Output Messages Notifications

CREATE INDEX

Query returned successfully in 88 msec.

202	SELECT AddRasterConstraints('schema_name'::name,
203	'tpi30_porto'::name,'rast'::name);
Data Output Messages Notifications	
<div> <div>+</div> <div>SQL</div> </div>	
	addrasterconstraints
	boolean
1	true

Algebra map

Przykład 1 - Wyrażenie Algebry Map

240	-- Przykład 1 - Wyrażenie Algebry Map
241	CREATE TABLE schema_name.porto_ndvi AS
242	WITH r AS (
243	SELECT a.rid,ST_Clip(a.rast, b.geom,true) AS rast
244	FROM rasters.landsat8 AS a, vectors.porto_parishes AS b
245	WHERE b.municipality ilike 'porto' and ST_Intersects(b.geom,a.rast)
246)
247	SELECT
248	r.rid,ST_MapAlgebra(
249	r.rast, 1,
250	r.rast, 4,
251	'([rast2.val] - [rast1.val]) / ([rast2.val] +
252	[rast1.val])::float','32BF'
253) AS rast
254	FROM r;
255	
256	SELECT st_summarystats(p.rast)
257	FROM schema_name.porto_ndvi AS p;
258	
Data Output Messages Notifications	
<div> <div>+</div> <div>SQL</div> </div>	
	st_summarystats
	summarystats
1	(1539,225.0319441196043,0.14621958682235497,0.10375721816385088,-0.05336048826575279,0.5319941639900208)
2	(1902,247.5309293824248,0.1301424444702549,0.06715705584959625,0.0012179126497358084,0.5395634770393372)
3	(434,49.67086953140097,0.11444900813686859,0.10111916069312085,-0.05224126577377319,0.46428191661834717)
4	(1311,312.8753030322114,0.23865393061190798,0.13831174716191197,-0.05492142215371132,0.5210691690444946)
5	(141,27.39425039291382,0.19428546377953063,0.07807146777039892,0.06713024526834488,0.3770729899406433)
6	(5376,1039.0867281809042,0.19328250152174556,0.10059616879015545,-0.01436083484441042,0.5354878902435303)
Total rows: 29 of 29 Query complete 00:00:00.088 Ln 256. Col 24	

Przykład 2 – Funkcja zwrotna

```
265 -- Przykład 2 - Funkcja zwrotna
266 CREATE OR REPLACE FUNCTION schema_name.ndvi(
267     VALUE double precision [] [] [],
268     pos integer [],
269     VARIADIC userargs text []
270 )
271 RETURNS double precision AS
272 $$
273 BEGIN
274     --RAISE NOTICE 'Pixel Value: %', value [1][1][1];-->For debug
275     purposes
276     RETURN (value [2][1][1] - value [1][1][1])/(value [2][1][1]+value [1][1][1]); --> NDVI calculation!
277 END;
278 $$
279 LANGUAGE 'plpgsql' IMMUTABLE COST 1000;
280
```

Data Output [Messages](#) Notifications

WARNING: there is no transaction in progress
COMMIT

Query returned successfully in 66 msec.

Przykład 3 - Funkcje TPI

```
281 -- Przykład 3 - Funkcje TPI
282 v CREATE TABLE schema_name.porto_ndvi2 AS
283 WITH r AS (
284 SELECT a.rid,ST_Clip(a.rast, b.geom,true) AS rast
285 FROM rasters.landsat8 AS a, vectors.porto_parishes AS b
286 WHERE b.municipality ilike 'porto' and ST_Intersects(b.geom,a.rast)
287 )
288 SELECT
289     r.rid,ST_MapAlgebra(
290     r.rast, ARRAY[1,4],
291     'schema_name.ndvi(double precision[],
292     integer[],text[])':::regprocedure, --> This is the function!
293     '32BF':::text
294 ) AS rast
295 FROM r;
296
297 SELECT * FROM schema_name.porto_ndvi2;
```

Data Output Messages Notifications



	rid integer	rast raster
1	427	01000001003849EE0BB84F3E404F2001E9AEB43DC075735DC2074FE3C02C78009F1B190441C
2	397	01000001003849EE0BB84F3E404F2001E9AEB43DC0076F60B09B56E3C075C249A272550441C
3	427	01000001003849EE0BB84F3E404F2001E9AEB43DC099C724180DBFE2C02C78009F1B190441C
4	428	01000001003849EE0BB84F3E404F2001E9AEB43DC01A7605D57B6FE2C02C78009F1B190441C
5	368	01000001003849EE0BB84F3E404F2001E9AEB43DC0D187F91C2C51E2C00224CA124B7E0441C
6	308	01000001003849EE0BB84F3E404F2001E9AEB43DC01A7605D57B6FE2C00B00B00A1E0750441C

Total rows: 29 of 29 Query complete 00:00:01.436 Ln 289, Col 23

```
299 -- Dodanie indeksu przestrzennego:
300 v CREATE INDEX idx_porto_ndvi2_rast_gist ON schema_name.porto_ndvi2
301 USING gist (ST_ConvexHull(rast));
302 -- Dodanie constraintów:
```

Data Output Messages Notifications

CREATE INDEX

Query returned successfully in 86 msec.


```

302 -- Dodanie constraintów:
303 SELECT AddRasterConstraints('schema_name'::name,
304 'porto_ndvi2'::name, 'rast'::name);


```

Data Output Messages Notifications

	addrasterconstraints	boolean
1	true	

Eksport danych

Przykład 0 - Użycie QGIS

 Edytuj połączenie z PostGIS

Informacja o połączeniu

Nazwa

aa

Usługa

Host

localhost

Port

5432

Baza danych

BDP6

Tryb SSL

wyłącz

Session ROLE

Uwierzytelnianie

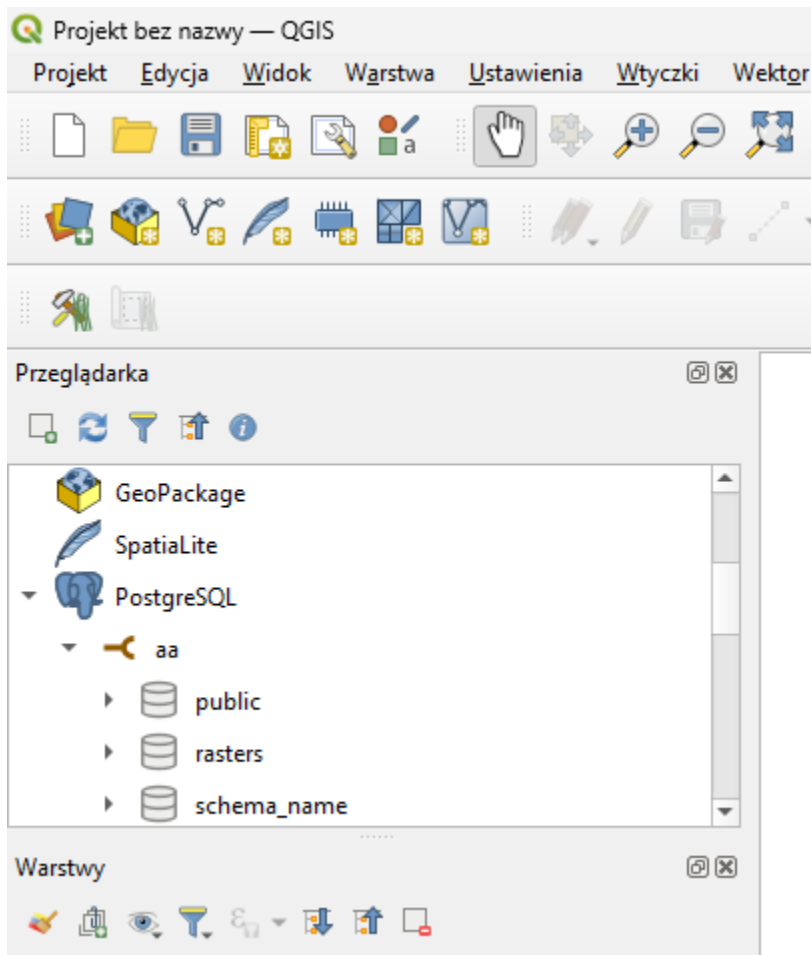
Konfiguracje

Bez zabezpieczeń

Wybierz lub utwórz konfigurację uwierzytelniania

hasło postgres (Basic)

Konfiguracja przechowuje zaszyfrowane dane w bazie danych uwierzytelniania QGIS.



Przykład 1 - ST_AsTiff

```

183 -- Eksport danych
184 -- Przykład 1 - ST_AsTiff
185
186 -- instalacja GDAL - https://trac.osgeo.org/osgeo4w/
187 -- przeniesienie binarek do postgresql/17/bin
188 SET postgis.gdal_enabled_drivers = 'ENABLE_ALL';
189
190 SELECT ST_AsTiff(ST_Union(rast))
191 FROM schema_name.porto_ndvi;
192

```

Data Output		Messages	Notifications
<div> <div>+</div> <div>📄</div> <div>▼</div> <div>📋</div> <div>▼</div> <div>🗑️</div> <div>🗄️</div> <div>⬇️</div> <div>📈</div> <div>SQL</div> </div>			
	st_astiff		
	bytea		
1	[binary dat...		

Przykład 2 - ST_AsGDALRaster

```
193 ▾ SELECT ST_AsGDALRaster(ST_Union(rast), 'GTiff', ARRAY['COMPRESS=DEFLATE',
194 'PREDICTOR=2', 'PZLEVEL=9'])
195 FROM schema_name.porto_ndvi;
196
197
```

Data Output Messages Notifications



	st_asgdalraster bytea
1	[binary data]

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

```
200 -- Przykład 3 - Zapisywanie danych na dysku za pomocą dużego obiektu (large object, lo)
201 ▾ CREATE TABLE tmp_out AS
202 SELECT lo_from_bytea(0,
203 ST_AsGDALRaster(ST_Union(rast), 'GTiff', ARRAY['COMPRESS=DEFLATE',
204 'PREDICTOR=2', 'PZLEVEL=9']))
205 ) AS loid
206 FROM schema_name.porto_ndvi;
207 -----
208 ▾ SELECT rolname, rolcanlogin, rolcreatorole, rolcreatedb, rolreplication, rolbypassrls
209 FROM pg_roles;
210 GRANT pg_write_server_files TO current_user;
211
212 -- cmd as admin -> icacls [redacted] /grant "NT AUTHORITY\NetworkService:(OI)(CI)F"
213 ▾ SELECT lo_export(loid, [redacted] 'c:\cwiczenia\bdp_cw6\temp.tiff')
214 FROM tmp_out;
215 -----
216 ▾ SELECT lo_unlink(loid)
217 FROM tmp_out; --> Delete the large object.
218
219
```

Data Output Messages Notifications



	lo_export integer
1	1

```
216 ▾ SELECT lo_unlink(loid)
217 FROM tmp_out; --> Delete the large object.
218
219
```






Data Output Messages Notifications



	lo_unlink integer
1	1

Przykład 4 - Użycie Gdal

```
C:\Program Files\OSGeo4W\bin>gdal_translate -sds -co COMPRESS=DEFLATE -co PREDICTOR=2 -co ZLEVEL=9 PG:"host=localhost port=5432 dbname=BDP6 user=postgres password=postgres schema=schema_name table=porto_ndvi" porto_ndvi.tiff
Input file size is 50, 44
0...10...20...30...40...50...60...70...80...90...100 - done.
Input file size is 47, 65
0...10...20...30...40...50...60...70...80...90...100 - done.
Input file size is 21, 34
0...10...20...30...40...50...60...70...80...90...100 - done.
Input file size is 64, 38
0...10...20...30...40...50...60...70...80...90...100 - done.
Input file size is 36, 9
0...10...20...30...40...50...60...70...80...90...100 - done.
```

 porto_ndvi_01	11/28/2024 12:44 PM	TIFF File	6 KB
 porto_ndvi_02	11/28/2024 12:44 PM	TIFF File	7 KB
 porto_ndvi_03	11/28/2024 12:44 PM	TIFF File	2 KB
 porto_ndvi_04	11/28/2024 12:44 PM	TIFF File	5 KB
 porto_ndvi_05	11/28/2024 12:44 PM	TIFF File	1 KB

Publikowanie danych za pomocą MapServer

Przykład 1 – Mapfile

```
Mar [redacted] PT MINGW64 ~/.vscode/docker-mapserver (master)
● $ docker cp c:/bdp6.map mapserver2:/
Successfully copied 2.56kB to mapserver2:/

Mar [redacted] PT MINGW64 ~/.vscode/docker-mapserver (master)
○ $ docker exec -it mapserver2 //bin//sh
# ls -l bdp*
-rwxr-xr-x 1 root root 678 Nov 28 17:03 bdp6.map
# ls -l *.png
-rw-r--r-- 1 www-data www-data 250439 Nov 28 16:42 map17328121522008.png
-rw-r--r-- 1 www-data www-data 250439 Nov 28 16:43 map17328122042009.png
-rw-r--r-- 1 www-data www-data 250439 Nov 28 16:59 map17328131652012.png
-rw-r--r-- 1 www-data www-data 250439 Nov 28 17:02 map17328133282021.png
-rw-r--r-- 1 www-data www-data 3010 Nov 28 17:05 map17328135022022.png
#
```

