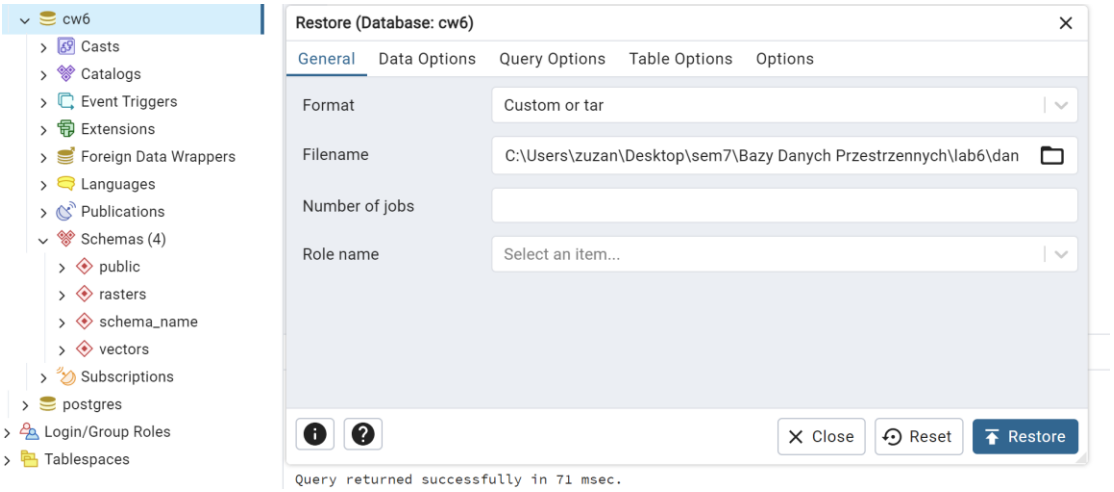
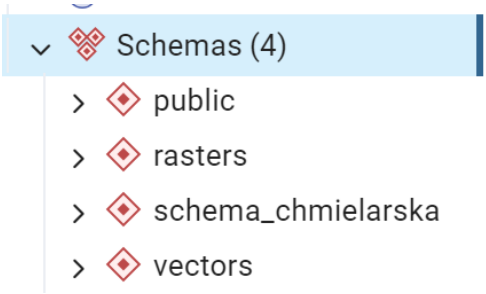
Nowa baza danych:



Struktura bazy danych:

ALTER SCHEMA schema_name RENAME TO schema_chmielarska;



Ładowanie danych rastrowych:

```
INSERT 0 1
CREATE INDEX
ANALYZE
UWAGA:  Adding SRID constraint
UWAGA:  Adding scale-X constraint
UWAGA:  Adding scale-Y constraint
UWAGA:  Adding blocksize-X constraint
UWAGA:  Adding blocksize-Y constraint
UWAGA:  Adding alignment constraint
UWAGA:  Adding number of bands constraint
UWAGA:  Adding pixel type constraint
UWAGA:  Adding nodata value constraint
UWAGA:  Adding out-of-database constraint
UWAGA:  Adding maximum extent constraint
 addrasterconstraints
----------------------
 t
(1 row)

COMMIT
VACUUM

C:\Program Files\PostgreSQL\16\bin>raster2pgsql.exe -s 3763 -N -32767 -t 100x100 -I -C -M -d "C:\Users\zuzan\Desktop\sem
7\Bazy Danych Przestrzennych\lab6\dane\srtm_1arc_v3.tif" rasters.dem | psql -d cw6 -h localhost -U postgres
```

```
COMMIT
VACUUM

C:\Program Files\PostgreSQL\16\bin>raster2pgsql.exe -s 3763 -N -32767 -t 128x128 -I -C -M -d "C:\Users\zuzan\Desktop\sem
7\Bazy Danych Przestrzennych\lab6\dane\Landsat8_L1TP_RGBN.tif" rasters.landsat8 | psql -d cw6 -h localhost -U postgres
```

```
4  SELECT * FROM public.raster_columns;
5
```

Data Output | Messages | Notifications

	r_table_catalog name	r_table_schema name	r_table_name name	r_raster_column name	srid integer
1	cw6	rasters	dem	rast	3763
2	cw6	rasters	landsat8	rast	3763

Tworzenie rastrów z istniejących rastrów i interakcja z wektorami:

ST_Intersects

```
--przecięcie rastra z wektorem
CREATE TABLE schema_chmielarska.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';

--klucz głowny
ALTER TABLE schema_chmielarska.intersects
ADD COLUMN rid SERIAL PRIMARY KEY;

--index przestrzenny GiST
CREATE INDEX idx_intersects_rast_gist ON schema_chmielarska.intersects
USING gist (ST_ConvexHull(rast));
--T_ConvexHull: przekształca raster na geometrię (otoczke wypukłą), bo gist działa na geom

-- schema::name table_name::name raster_column::name --true/false
--Constraint (ograniczenie) dla typu danych: Sprawdza, czy kolumna zawiera poprawne dane r
--także czy wartości w tej kolumnie są poprawnie zdefiniowane
SELECT AddRasterConstraints('schema_chmielarska'::name,
'intersects'::name,'rast'::name);

SELECT * FROM schema_chmielarska.intersects;
```

Data Output | Messages | Notifications

	addrasterconstraints boolean
1	true

ST_Clip

```
----
--ST_Clip-przycina raster, zostawiając tylko te jego części, które pokrywają się z obszarem geometrii
CREATE TABLE schema_chmielarska.clip AS
SELECT ST_Clip(a.rast, b.geom, true) AS rast, b.municipality
FROM rasters.dem AS a, vectors.porto_parishes AS b
WHERE ST_Intersects(a.rast, b.geom) AND b.municipality like 'PORTO';

ALTER TABLE schema_chmielarska.clip
ADD COLUMN rid SERIAL PRIMARY KEY;

CREATE INDEX idx_clip_rast_gist ON schema_chmielarska.clip
USING gist (ST_ConvexHull(rast));

SELECT AddRasterConstraints('schema_chmielarska'::name,
'clip'::name,'rast'::name);

SELECT * FROM schema_chmielarska.clip;
```

Output | Messages | Notifications

addrasterconstraints boolean
true

ST_Union

```
--ST_Union
CREATE TABLE schema_chmielarska.union AS
SELECT ST_Union(ST_Clip(a.rast, b.geom, true)) AS rast --true -> piksele spoza obszaru wycinka są wypełnione NUL
FROM rasters.dem AS a, vectors.porto_parishes AS b
WHERE b.municipality ILIKE 'porto' and ST_Intersects(b.geom,a.rast);
--połączenie wyciętych fragmentów

ALTER TABLE schema_chmielarska.union
ADD COLUMN rid SERIAL PRIMARY KEY;

CREATE INDEX idx_union_rast_gist ON schema_chmielarska.union
USING gist (ST_ConvexHull(rast));

SELECT AddRasterConstraints('schema_chmielarska'::name,
'union'::name,'rast'::name);

SELECT * FROM schema_chmielarska.union;

--przyklad
SELECT ST_Union(rast, 'MEAN') --default: LAST
FROM schema_chmielarska.clip;
```

Data Output | Messages | Notifications

	st_union raster
1	01000001006172BF3E4D5A374080318D6907CA3EC0D6994F2F0BE0E6C0DAE35DC00896044100000000000000000000000000000000B30E0000F201AC0045018001800180

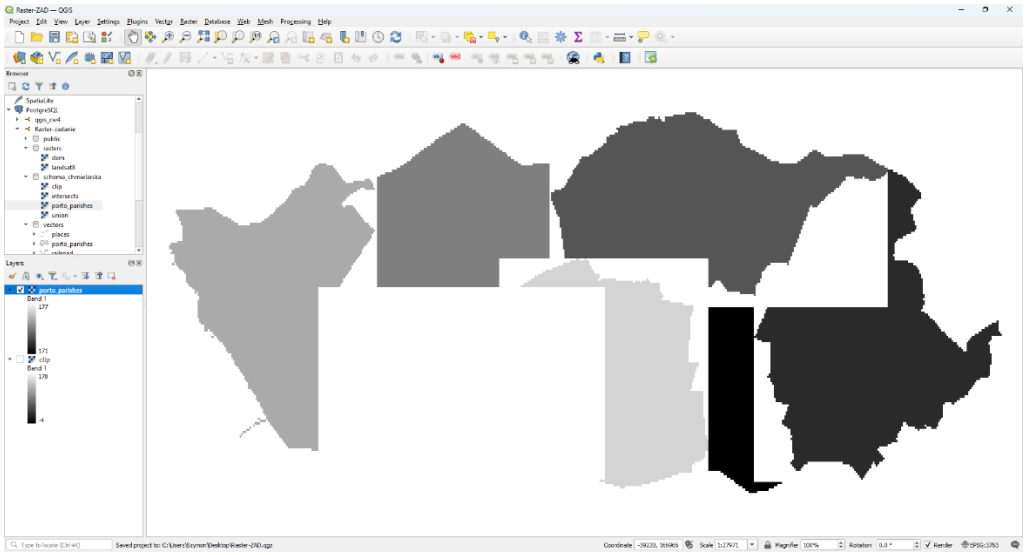
Tworzenie rastrów z wektorów (rastrowanie):

ST_AsRaster

```
----
--ST_AsRaster
CREATE TABLE schema_chmielarska.porto_parishes AS
WITH r AS (
    SELECT rast FROM rasters.dem
    LIMIT 1
)
--pobieramy wzorcowy raster, aby powstałe rastry miały tę samą rozdzielczość i układ współrzędnych
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';


ALTER TABLE schema_chmielarska.porto_parishes
ADD COLUMN rid SERIAL PRIMARY KEY;

CREATE INDEX idx_porto_parishes_rast_gist ON schema_chmielarska.porto_parishes
USING gist (ST_ConvexHull(rast));

SELECT AddRasterConstraints('schema_chmielarska'::name,
'porto_parishes'::name,'rast'::name);
```

Data Output | Messages | Notifications

	rast raster
1	01000001006172BF3E4D5A374080318D6907CA3EC0574768B43454E3C0D0979D709B5404410000000000000000000000000000000B30E000042006900440000000000000

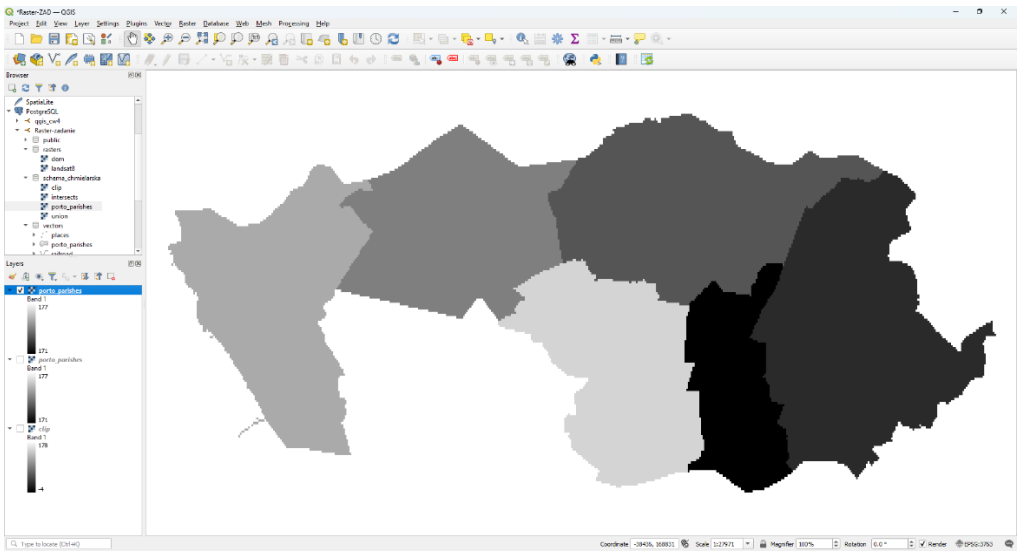


ST_Union

```sql
--ST_Union
--łączymu rekordy z poprzedniego przykładu w pojedynczy raster
DROP TABLE schema_chmielarska.porto_parishes; --> drop table porto_parishes first
CREATE TABLE schema_chmielarska.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';

ALTER TABLE schema_chmielarska.porto_parishes
ADD COLUMN rid SERIAL PRIMARY KEY;

CREATE INDEX idx_porto_parishes_rast_gist ON schema_chmielarska.porto_parishes
USING gist (ST_ConvexHull(rast));

SELECT AddRasterConstraints('schema_chmielarska'::name,
'porto_parishes'::name,'rast'::name);

SELECT * FROM schema_chmielarska.porto_parishes;

COPY (
    SELECT ST_AsTIFF(rast)
    FROM schema_chmielarska.porto_parishes
) TO 'C:/Users/zuzan/Desktop/sem7/Bazy Danych Przestrzennych/lab6/porto_parishes_after.tif'

```

Data Output | Messages | Notifications

```
COPY 1
```



ST_Tile

```
131  ----
132  --ST_Tile--podział rastra na kafelki
133  DROP TABLE schema_chmielarska.porto_parishes; --> drop table porto_parishes first
134  CREATE TABLE schema_chmielarska.porto_parishes AS
135  WITH r AS (
136      SELECT rast FROM rasters.dem
137      LIMIT 1
138  )--łączymy w jeden raster i dzielimy na kafelki 128x128
139  SELECT ST_Tile(ST_Union(ST_AsRaster(a.geom,r.rast,'8BUI',a.id,-32767)),128,128,true,-32767) AS rast
140  FROM vectors.porto_parishes AS a, r
141  WHERE a.municipality ILIKE 'porto';
142
143  ALTER TABLE schema_chmielarska.porto_parishes
144  ADD COLUMN rid SERIAL PRIMARY KEY;
145
146  CREATE INDEX idx_porto_parishes_rast_gist ON schema_chmielarska.porto_parishes
147  USING gist (ST_ConvexHull(rast));
148
149  SELECT AddRasterConstraints('schema_chmielarska'::name,
150  'porto_parishes'::name,'rast'::name);
151
152  SELECT * FROM schema_chmielarska.porto_parishes;
```

Data Output | Messages | Notifications

	rast raster
1	01000001006172BF3E4D5A374080318D6907CA3EC0D6994F2F0BE0E6C0DAE35DC008960441000000000000000000000000000000B30E00008000800044000000000C
2	01000001006172BF3E4D5A374080318D6907CA3EC0B0A2635B666AE5C0DAE35DC008960441000000000000000000000000000000B30E00008000800044000000000C

Konwertowanie rastrów na wektory (wektoryzowanie):

ST_Intersection

```
168  '''
169  CREATE TABLE schema_chmielarska.intersection AS
170  SELECT a.rid,(ST_Intersection(b.geom,a.rast)).geom,(ST_Intersection(b.geom,a.rast)).val
171  FROM rasters.landsat8 AS a, vectors.porto_parishes AS b
172  WHERE b.parish ILIKE 'paranhos' AND ST_Intersects(b.geom,a.rast);
173
174  SELECT * FROM schema_chmielarska.intersection;
```

Data Output | Messages | Notifications

	rid integer	geom geometry
1	221	0103000020B30E0000010000000500000086F60B09B56E3C0E32C703B809504410AB086D0325AE3C0E32C703B8095044179
2	221	0103000020B30E0000010000000500000003EF15EB9D152E3C0E32C703B80950441086F60B09B56E3C0E32C703B8095044108
3	221	0103000020B30E0000010000000400000BF61697A576DF3C0DAF4F895929404410EDB00034E6EF3C0DAF4F895929404418F

ST_DumpAsPolygons

```
176  ----
177  --ST_DumpAsPolygons->konwertuje rastry w wektory (poligony)
178  CREATE TABLE schema_chmielarska.dumppolygons AS
179  SELECT a.rid,(ST_DumpAsPolygons(ST_Clip(a.rast,b.geom))).geom,(ST_DumpAsPolygons(ST_Clip(a.rast,b.geom))).val
180  FROM rasters.landsat8 AS a, vectors.porto_parishes AS b
181  WHERE b.parish ILIKE 'paranhos' AND ST_Intersects(b.geom,a.rast);
182
183  SELECT rid, ST_AsText(geom), val FROM schema_chmielarska.dumppolygons;
```

Data Output | Messages | Notifications

	rid integer	st_astext text
1	221	POLYGON((-39665.488084514815 168624.02902255123,-39665.488084514815 168564.61743471635,-39635.17668243649 168564.61743471635,-39635.1766
2	221	POLYGON((-39635.17668243649 168624.02902255123,-39635.17668243649 168564.61743471635,-39604.86528035817 168564.61743471635,-39604.865280
3	221	POLYGON((-39604.86528035817 168624.02902255123,-39604.86528035817 168564.61743471635,-39574.553878279854 168564.61743471635,-39574.55387

Analiza rastrów:

ST_Band

```
186  --ST_Band
187  CREATE TABLE schema_chmielarska.landsat_nir AS
188  SELECT rid, ST_Band(rast,4) AS rast --wyodrębnia 4 pasmo z obrazu rastrowego
189  FROM rasters.landsat8;
190
191  SELECT * FROM schema_chmielarska.landsat_nir
192
```

Data Output | Messages | Notifications

	rid integer	rast raster
1	1	01000001003849EE0BB84F3E404F2001E9AEB43DC02068193DBBCBECC071695283BC2C094100000000000000000000000000000000B30E000080008000468
2	2	01000001003849EE0BB84F3E404F2001E9AEB43DC08C835ABCBFE6EAC071695283BC2C094100000000000000000000000000000000B30E000080008000468

ST_Clip

```
193  ----
194  --ST_Clip--tutaj używany do wycięcia rastra z innego rastra, przycięcie rastra do granic geometrii paranhos
195  CREATE TABLE schema_chmielarska.paranhos_dem AS
196  SELECT a.rid,ST_Clip(a.rast, b.geom,true) as rast
197  FROM rasters.dem AS a, vectors.porto_parishes AS b
198  WHERE b.parish ILIKE 'paranhos' AND ST_Intersects(b.geom,a.rast);
199
200  CREATE INDEX idx_paranhos_dem_rast_gist ON schema_chmielarska.paranhos_dem
201  USING gist (ST_ConvexHull(rast));
202
203  SELECT AddRasterConstraints('schema_chmielarska'::name,
204  'paranhos_dem'::name,'rast'::name);
205
206  SELECT * FROM schema_chmielarska.paranhos_dem;
207
```

Data Output | Messages | Notifications

	rid integer	rast raster
1	380	01000001006172BF3E4D5A374080318D6907CA3EC09A49D3957D46E4C033B2707F2F92044100000000000000000000000000000000B30E000035004900450
2	382	01000001006172BF3E4D5A374080318D6907CA3EC02F3C8390DE87F2C0D7D06D6CAD85044100000000000000000000000000000000B30E000031002000450

ST_Slope

```
208  ----
209  --ST_Slope-oblicza nachylenie na podstawie różnicy wysokości między pikselami w sąsiedztwie
210  CREATE TABLE schema_chmielarska.paranhos_slope AS
211  SELECT a.rid,ST_Slope(a.rast,1,'32BF','PERCENTAGE') AS rast
212  FROM schema_chmielarska.paranhos_dem AS a;
213
214  CREATE INDEX idx_paranhos_slope_rast_gist ON schema_chmielarska.paranhos_slope
215  USING gist (ST_ConvexHull(rast));
216
217  SELECT AddRasterConstraints('schema_chmielarska'::name,
218  'paranhos_slope'::name,'rast'::name);
219
220  SELECT * FROM schema_chmielarska.paranhos_slope;
```

Data Output | Messages | Notifications

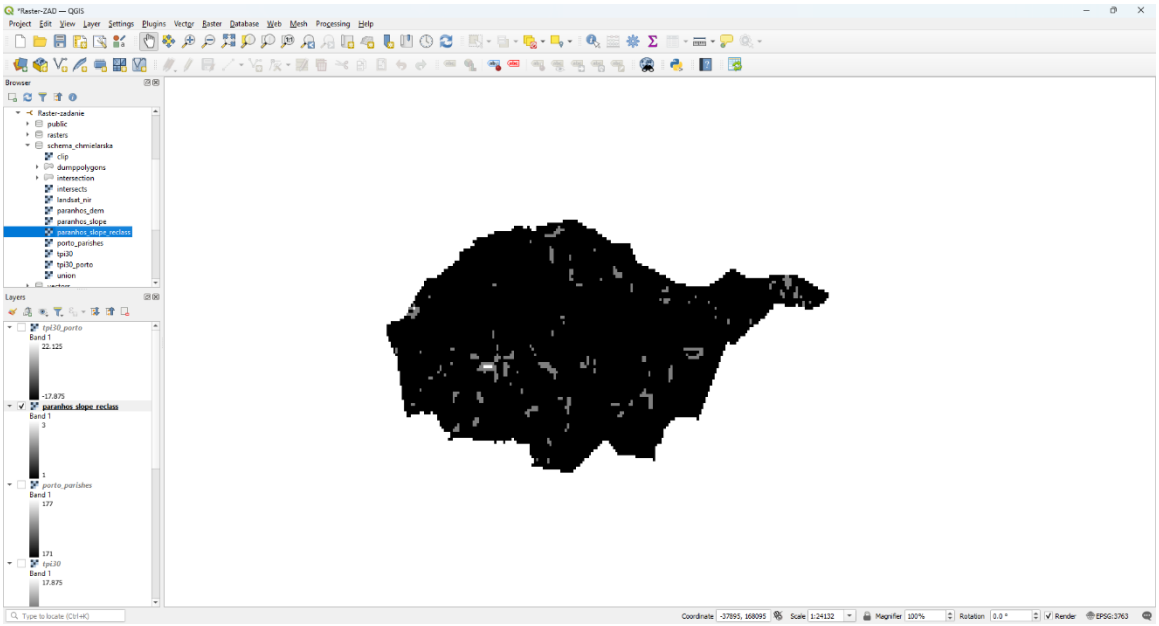
	rid integer	rast raster
1	380	01000001006172BF3E4D5A374080318D6907CA3EC09A49D3957D46E4C033B2707F2F920441000000000000000000000000000000B30E0000350049004A0(
2	382	01000001006172BF3E4D5A374080318D6907CA3EC02E3C8390DE87E2C0D7D06D6CAD850441000000000000000000000000000000B30E0000310020004A(
3	412	01000001006172BF3E4D5A374080318D6907CA3EC0187635E2BF88E3C0474F11FE054A044100000000000000000000000000000B30E00003B0008004A0(

ST_Reclass

```
223  --ST_Reclass--podział terenu na kategorie według % slope
224  CREATE TABLE schema_chmielarska.paranhos_slope_reclass AS
225  SELECT a.rid,ST_Reclass(a.rast,1,']0-15]:1, (15-30]:2, (30-9999:3','32BF',0) AS rast
226  ----------------------raster, pasmo, przedzialy
227  FROM schema_chmielarska.paranhos_slope AS a;
228
229  CREATE INDEX idx_paranhos_slope_reclass_rast_gist ON schema_chmielarska.paranhos_slope_reclass
230  USING gist (ST_ConvexHull(rast));
231
232  SELECT AddRasterConstraints('schema_chmielarska'::name,
233  'paranhos_slope_reclass'::name,'rast'::name);
234
235  SELECT * FROM schema_chmielarska.paranhos_slope_reclass;
236  ----
```

Data Output | Messages | Notifications

	rid integer	rast raster
1	380	01000001006172BF3E4D5A374080318D6907CA3EC09A49D3957D46E4C033B2707F2F920441000000000000000000000000000000B30E0000350049004A0
2	382	01000001006172BF3E4D5A374080318D6907CA3EC02E3C8390DE87E2C0D7D06D6CAD850441000000000000000000000000000000B30E0000310020004A



ST_SummaryStats

```
238  --ST_SummaryStat
239  SELECT ST_SummaryStats(a.rast) AS stats
240  FROM schema_chmielarska.paranhos_dem AS a;
241  --count,sum,mean,stddev,min,max
242
```

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)

ST_SummaryStats oraz Union

```
244  --ST_SummaryStats oraz Union
245  SELECT st_summarystats(ST_Union(a.rast))
246  FROM schema_chmielarska.paranhos_dem AS a;
247
```

Data Output | Messages | Notifications

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

ST_SummaryStats z lepszą kontrolą złożonego typu danych

```
250  --ST_SummaryStats z lepszą kontrolą złożonego typu danych
251  WITH t AS (
252      SELECT st_summarystats(ST_Union(a.rast)) AS stats
253      FROM schema_chmielarska.paranhos_dem AS a
254  )
255  SELECT (stats).count,(stats).sum,(stats).mean,(stats).stddev,(stats).min,(stats).max FROM t
```

Data Output | Messages | Notifications

	count bigint	sum double precision	mean double precision	stddev double precision	min double precision	max double precision
1	9977	1222455	122.52731281948482	16.908004202736272	87	158

ST_SummaryStats w połączeniu z GROUP BY

```
--ST_SummaryStats w połączeniu z GROUP BY
--statystyki z podziałem na parish
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;
```

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

ST_Value

```
--jednopunktowej, należy przekonwertować geometrię wielopunktową na geometr
--za pomocą funkcji (ST_Dump(b.geom)).geom.
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;
```

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

ST_TPI

```
--ST_TPI
--ST_Value pozwala na utworzenie mapy TPI z DEM wysokości
CREATE TABLE schema_chmielarska.tpi30 AS
SELECT ST_TPI(a.rast,1) AS rast
FROM rasters.dem a;
```

Data Output | Messages | Notifications

```
SELECT 589

Query returned successfully in 1 min 14 secs.
```

```sql
--ST_TPI dla Porto
CREATE TABLE schema_chmielarska.tpi30_porto 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_porto_rast_gist ON schema_chmielarska.tpi30_porto
USING gist (ST_ConvexHull(rast));

SELECT AddRasterConstraints('schema_chmielarska'::name,
'tpi30_porto'::name,'rast'::name);

COPY (
    SELECT ST_AsTIFF(rast)
    FROM schema_chmielarska.tpi30_porto
) TO 'C:/Users/zuzan/Desktop/sem7/Bazy Danych Przestrzennych/lab6/tpi30_porto.tif'

```

Data Output | Messages | Notifications

```
SELECT 25

Query returned successfully in 2 secs 756 msec.
```

Wyrażenie Algebry Map:

```sql
--NDVI=(NIR-Red)/(NIR+Red)
--NDVI (Normalized Difference Vegetation Index)
CREATE TABLE schema_chmielarska.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 schema_chmielarska.porto_ndvi
USING gist (ST_ConvexHull(rast));

SELECT AddRasterConstraints('schema_chmielarska'::name,
'porto_ndvi'::name,'rast'::name);

SELECT * FROM schema_chmielarska.porto_ndvi;
```

Data Output | Messages | Notifications

	rid integer	rast raster
1	245	01000001003849EE0BB84F3E404F2001E9AEB43DC0086F60B09B56E3C01F1195122C320441000000000000000000000000000000000B30E0
2	270	01000001003849EE0BB84F3E404F2001E9AEB43DC075735DC2074EE3C0E4374673AE1104410000000000000000000000000000000B30E0

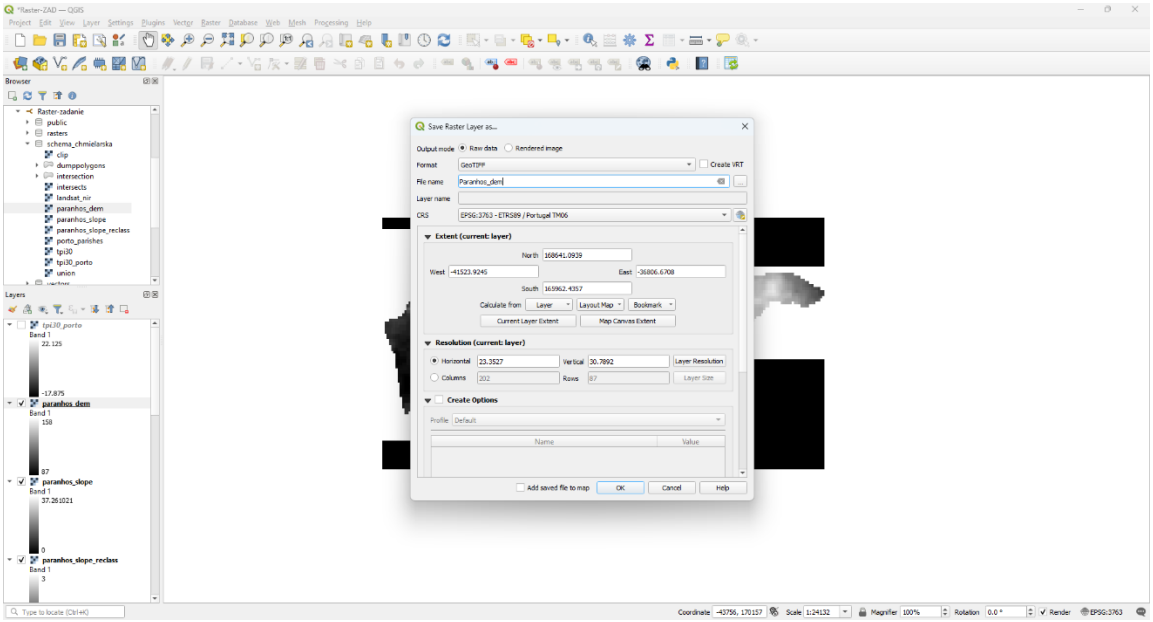
```sql
--wywołanie funkcji w query
CREATE TABLE schema_chmielarska.porto_ndvi2 AS
WITH r AS (
    SELECT a.rid,ST_Clip(a.rast, b.geom,true) AS rast
    FROM rasters.landsat8 AS a, vectors.porto_parishes AS b
    WHERE b.municipality ilike 'porto' and ST_Intersects(b.geom,a.rast)
)
SELECT r.rid,ST_MapAlgebra(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

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

SELECT AddRasterConstraints('schema_chmielarska'::name,
'porto_ndvi2'::name,'rast'::name);

SELECT * FROM schema_chmielarska.porto_ndvi2;
```

Data Output | Messages | Notifications

	rid integer	rast raster
1	245	01000001003849EE0BB84F3E404F2001E9AEB43DC0086F60B09B56E3C01F1195122C320441000000000000000000000000000000B30E0000
2	270	01000001003849EE0BB84F3E404F2001E9AEB43DC075735DC2074FE3C0E4374673AE110441000000000000000000000000000000B30E0000

Eksport danych:

Użycie QGIS:



ST_AsTiff:

```
381  --ST_AsTiff
382  SELECT ST_AsTiff(ST_Union(rast))
383  FROM schema_chmielarska.porto_ndvi;
384
385
386
```

Data Output | Messages | Notifications

	st_astiff bytea
1	[binary dat...

ST_AsGDALRaster

```
385  --ST_AsGDALRaster
386  SELECT ST_AsGDALRaster(ST_Union(rast), 'GTiff', ARRAY['COMPRESS=DEFLATE','PREDICTOR=2', 'PZLEVEL=9'])
387  FROM schema_chmielarska.porto_ndvi;
388
```

Data Output | Messages | Notifications

	st_asgdalraster bytea
1	[binary data]

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

```
369  --Large object
370  CREATE TABLE tmp_out AS
371  SELECT lo_from_bytea(0,
372  ST_AsGDALRaster(ST_Union(rast), 'GTiff', ARRAY['COMPRESS=DEFLATE','PREDICTOR=2', 'PZLEVEL=9'])) AS loid
373  FROM schema_chmielarska.porto_ndvi;
374  ----------------------------------------------
375  SELECT lo_export(loid, 'C:\Users\zuzan\Desktop\sem7\Bazy Danych Przestrzennych\lab6\myraster.tiff') --> Save
376  FROM tmp_out;
377  ----------------------------------------------
378  SELECT lo_unlink(loid)
379  FROM tmp_out; --> Delete the large object.
```

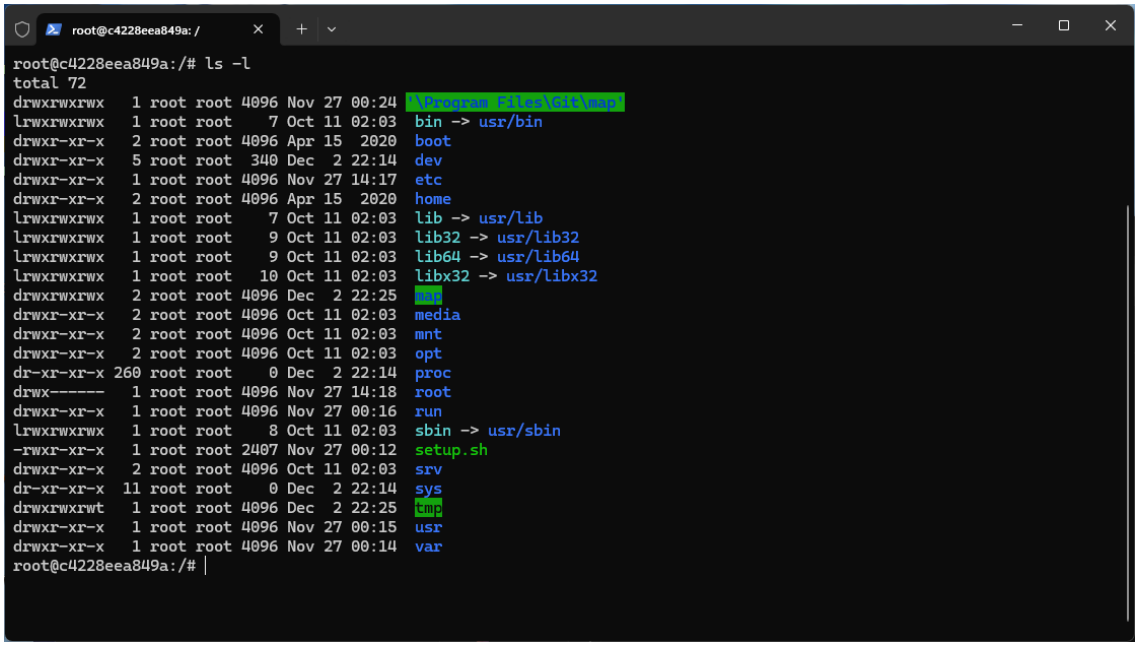
Data Output | Messages | Notifications

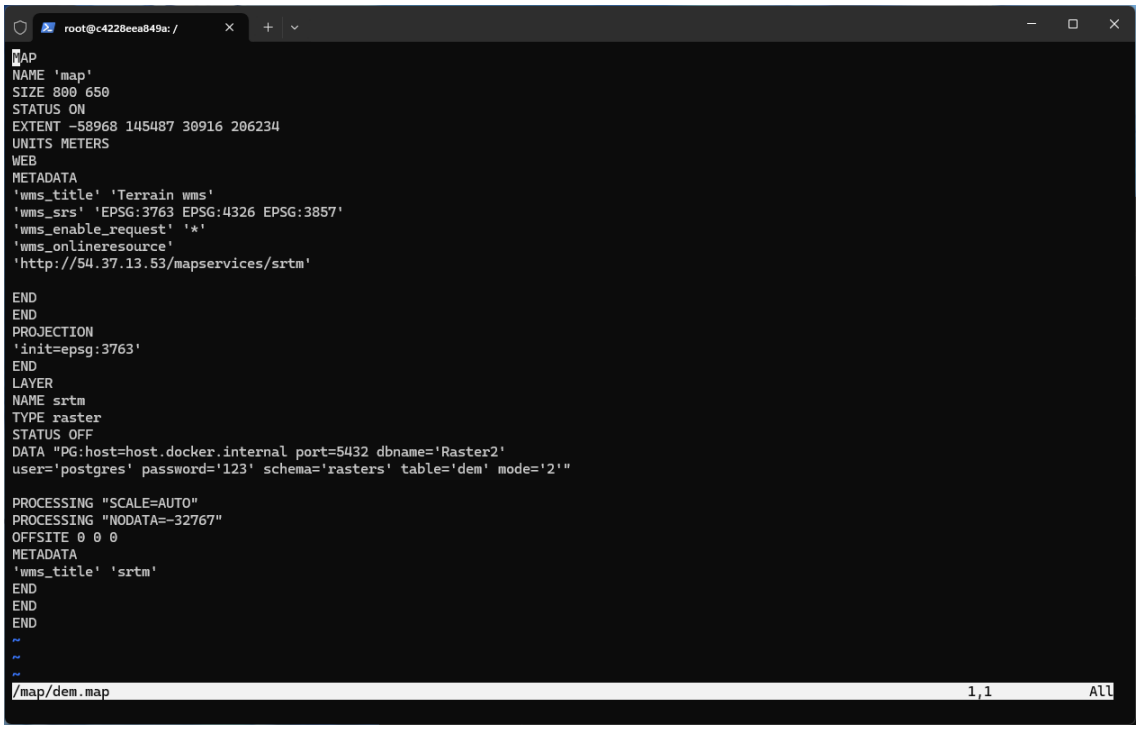
	lo_unlink integer
1	1

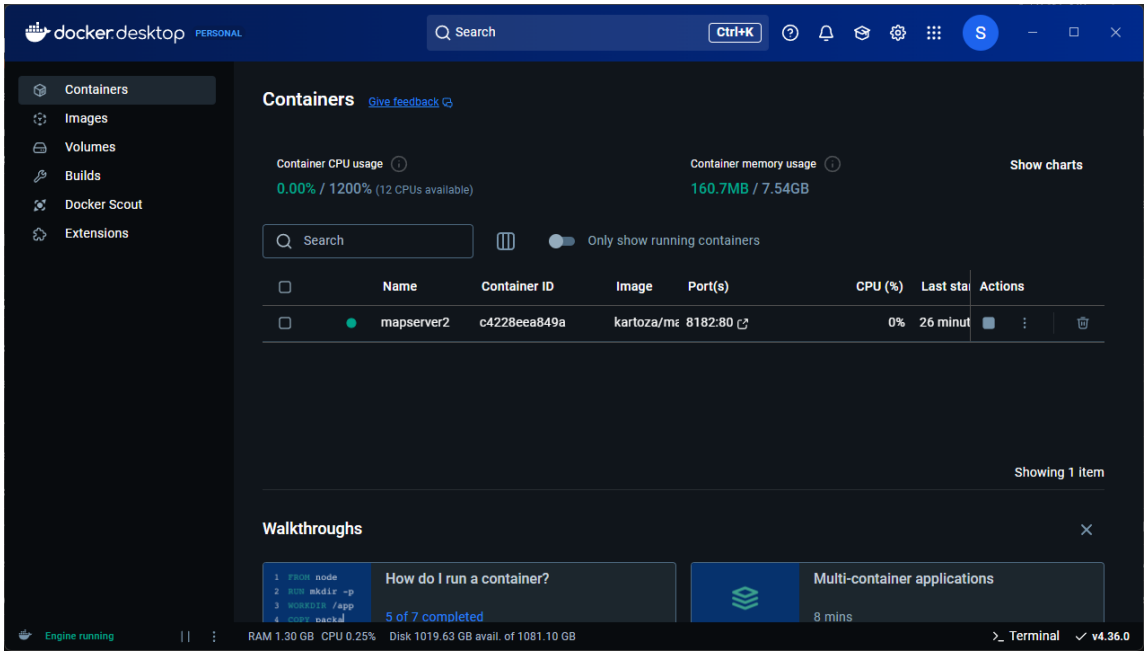
Użycie Gdal

```
C:\Users\zuzan\Desktop\sem7\Bazy Danych Przestrzennych\lab6>gdal_translate -co COMPRESS=DEFLATE -co PREDICTOR=2 -co ZLEV
EL=9 PG:"host=localhost port=5432 dbname=cw6 user=postgres password=123 schema=schema_chmielarska table=porto_ndvi mode=
2" porto_ndvi.tiff
```

MapServer


root@c4228eea849a: /
root@c4228eea849a:/# ls -l
total 72
drwxrwxrwx   1 root root 4096 Nov 27 00:24 'C:\Program Files\Git\map'
lrwxrwxrwx   1 root root   7 Oct 11 02:03 bin -> usr/bin
drwxr-xr-x   2 root root 4096 Apr 15 2020 boot
drwxr-xr-x   5 root root 340 Dec 2 22:14 dev
drwxr-xr-x   1 root root 4096 Nov 27 14:17 etc
drwxr-xr-x   2 root root 4096 Apr 15 2020 home
lrwxrwxrwx   1 root root   7 Oct 11 02:03 lib -> usr/lib
lrwxrwxrwx   1 root root   9 Oct 11 02:03 lib32 -> usr/lib32
lrwxrwxrwx   1 root root   9 Oct 11 02:03 lib64 -> usr/lib64
lrwxrwxrwx   1 root root   10 Oct 11 02:03 libx32 -> usr/libx32
drwxrwxrwx   2 root root 4096 Dec 2 22:25 map
drwxr-xr-x   2 root root 4096 Oct 11 02:03 media
drwxr-xr-x   2 root root 4096 Oct 11 02:03 mnt
drwxr-xr-x   2 root root 4096 Oct 11 02:03 opt
dr-xr-xr-x 260 root root   0 Dec 2 22:14 proc
drwx------   1 root root 4096 Nov 27 14:18 root
drwxr-xr-x   1 root root 4096 Nov 27 00:16 run
lrwxrwxrwx   1 root root   8 Oct 11 02:03 sbin -> usr/sbin
-rwxr-xr-x   1 root root 2407 Nov 27 00:12 setup.sh
drwxr-xr-x   2 root root 4096 Oct 11 02:03 srv
dr-xr-xr-x 11 root root   0 Dec 2 22:14 sys
drwxrwxrwt   1 root root 4096 Dec 2 22:25 tmp
drwxr-xr-x   1 root root 4096 Nov 27 00:15 usr
drwxr-xr-x   1 root root 4096 Nov 27 00:14 var
root@c4228eea849a:/#


root@c4228eea849a: /
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='Raster2'
user='postgres' password='123' schema='rasters' table='dem' mode='2'"
PROCESSING "SCALE=AUTO"
PROCESSING "NODATA=-32767"
OFFSITE 0 0 0
METADATA
'wms_title' 'srtm'
END
END
END
/map/dem.map 1,1 All



Nie udało się wyświetlić mapy...