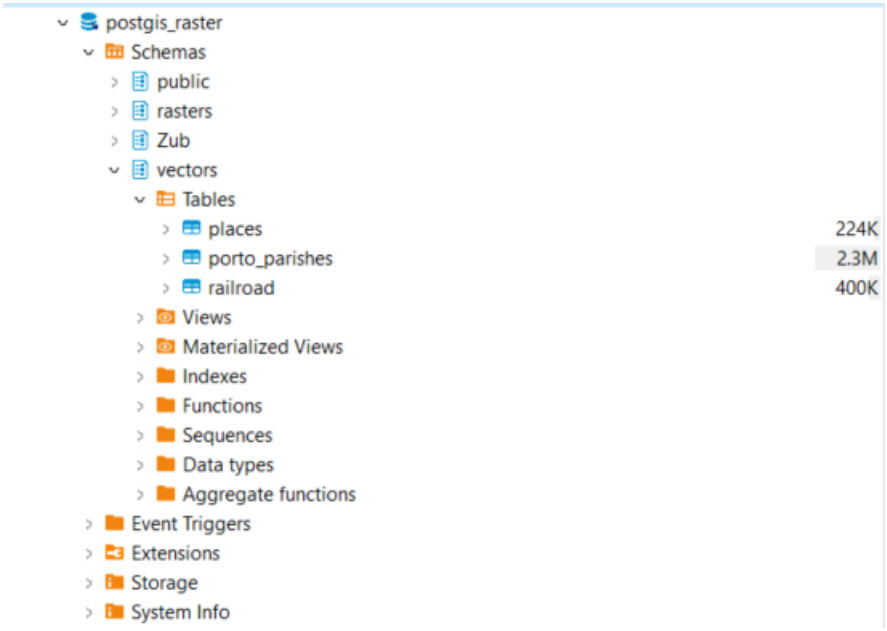
Nowa baza danych

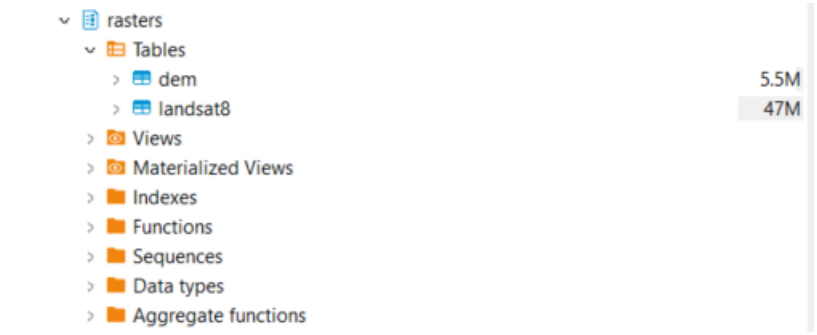


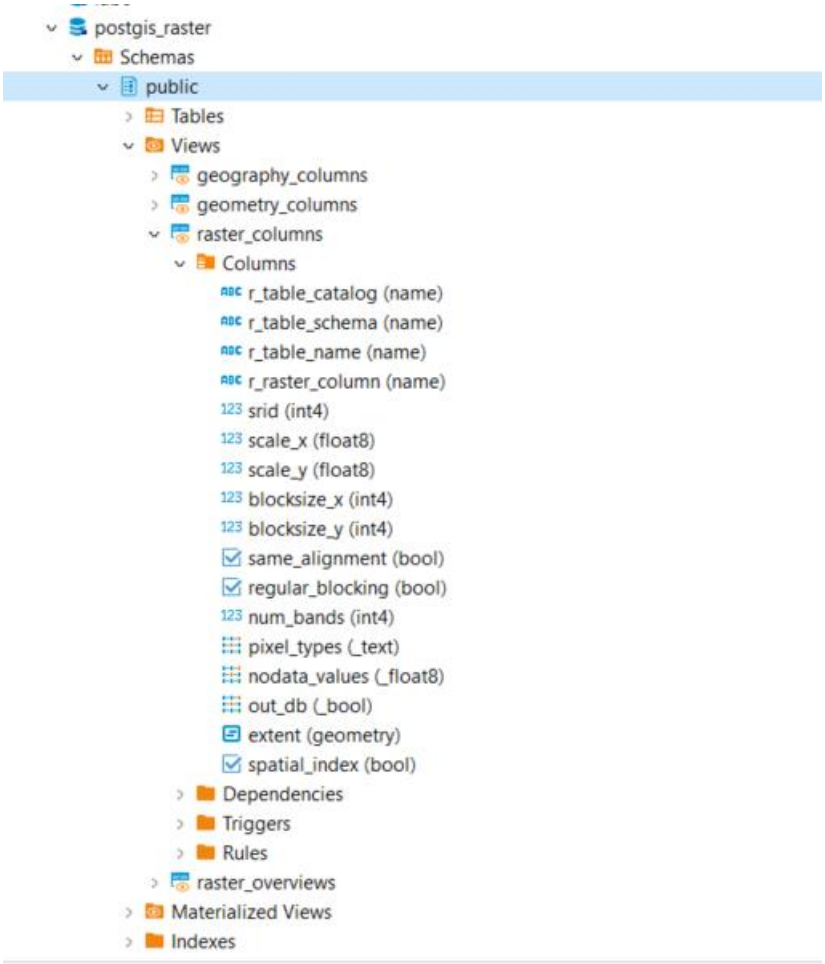
Ładowanie danych rastrowych

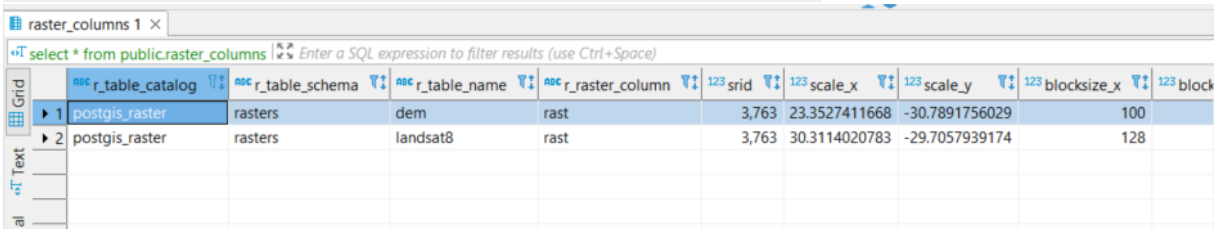
```
D:\Program Files\PostgreSQL\14\bin>raster2pgsql.exe -s 3763 -N -32767 -t 100x100 -I -C -M -d D:\SQL_code\lab_7\srtm_1arc
_v3.tif rasters.dem > D:\SQL_code\lab_7\dem.sql
Processing 1/1: D:\SQL_code\lab_7\srtm_1arc_v3.tif
```

```
D:\Program Files\PostgreSQL\14\bin>raster2pgsql.exe -s 3763 -N -32767 -t 100x100 -I -C -M -d D:\SQL_code\lab_7\srtm_1arc
_v3.tif rasters.dem | psql -d postgis_raster -h localhost -U postgres -p 5432
Password for user postgres: Processing 1/1: D:\SQL_code\lab_7\srtm_1arc_v3.tif

BEGIN
NOTICE:  table "dem" does not exist, skipping
DROP TABLE
CREATE TABLE
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
```

```
D:\Program Files\PostgreSQL\14\bin>raster2pgsql.exe -s 3763 -N -32767 -t 128x128 -I -C -M -d D:\SQL_code\lab_7\Landsat8_
L1TP_RGBN.tif rasters.landsat8 | psql -d postgis_raster -h localhost -U postgres -p 5432
Processing 1/1: D:\SQL_code\lab_7\Landsat8_L1TP_RGBN.tif
Password for user postgres:
BEGIN
NOTICE:  table "landsat8" does not exist, skipping
DROP TABLE
CREATE TABLE
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
```

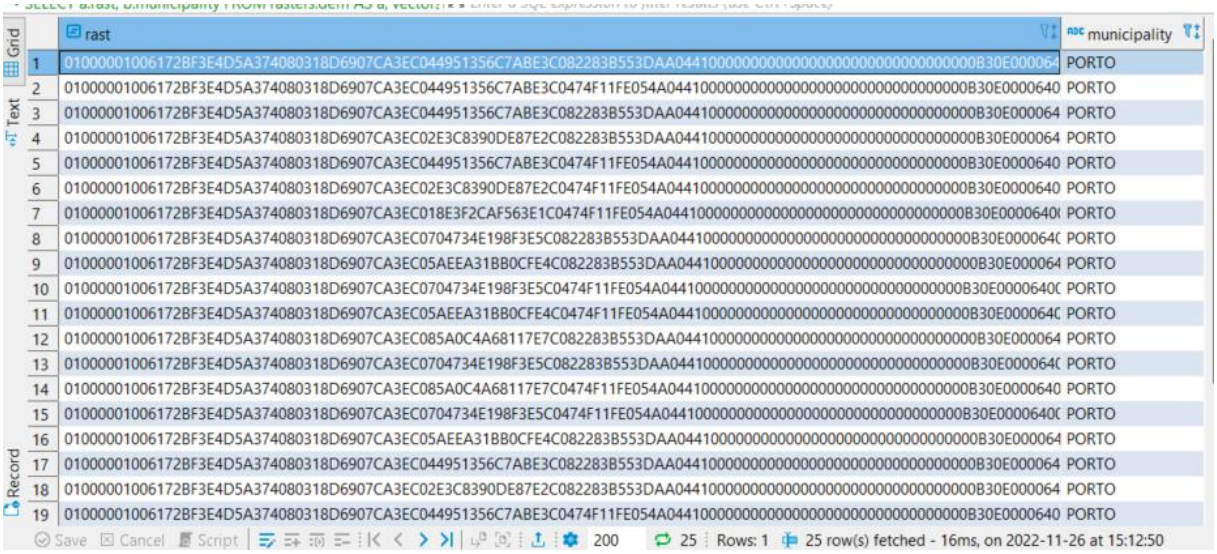




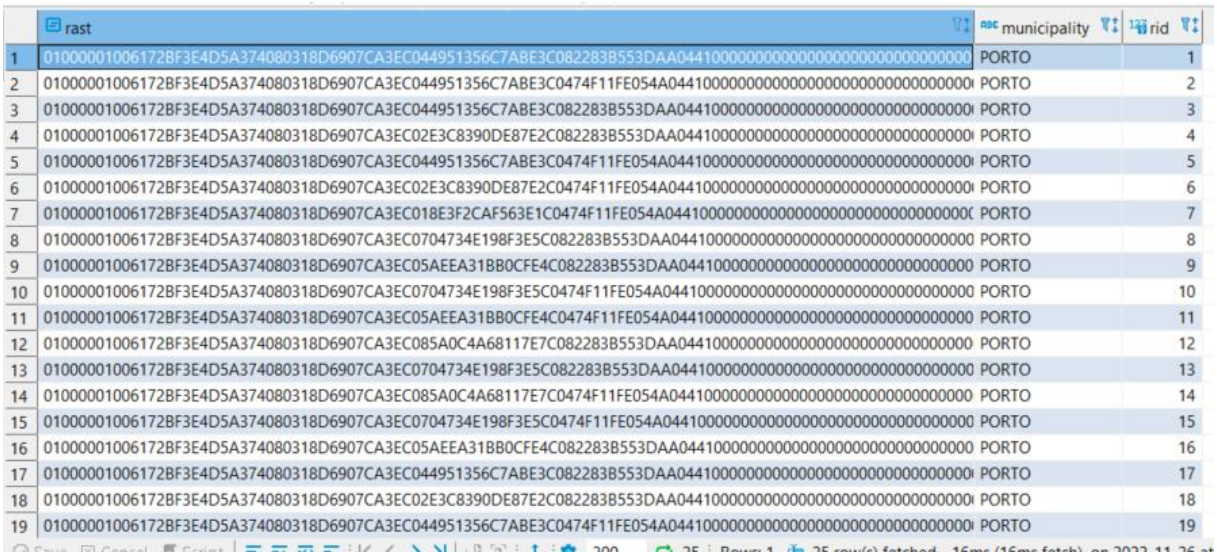


Tworzenie rastrów z istniejących rastrów i interakcja z wektorami

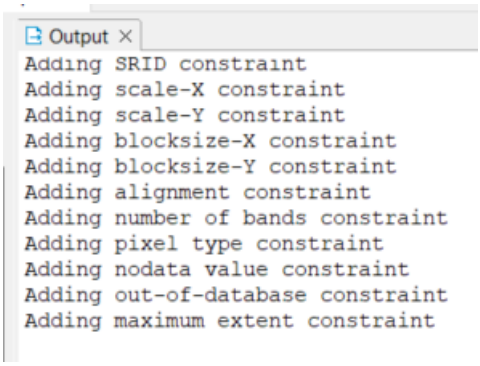
-- 1 przykład



	rast	municipality
1	01000001006172BF3E4D5A374080318D6907CA3EC044951356C7ABE3C082283B553DAA0441000000000000000000000000000000B30E000064	PORTO
2	01000001006172BF3E4D5A374080318D6907CA3EC044951356C7ABE3C0474F11FE054A0441000000000000000000000000000000B30E0000640	PORTO
3	01000001006172BF3E4D5A374080318D6907CA3EC044951356C7ABE3C082283B553DAA0441000000000000000000000000000000B30E000064	PORTO
4	01000001006172BF3E4D5A374080318D6907CA3EC02E3C8390DE87E2C082283B553DAA0441000000000000000000000000000000B30E000064	PORTO
5	01000001006172BF3E4D5A374080318D6907CA3EC044951356C7ABE3C0474F11FE054A0441000000000000000000000000000000B30E0000640	PORTO
6	01000001006172BF3E4D5A374080318D6907CA3EC02E3C8390DE87E2C0474F11FE054A0441000000000000000000000000000000B30E0000640	PORTO
7	01000001006172BF3E4D5A374080318D6907CA3EC018E3F2CAF563E1C0474F11FE054A0441000000000000000000000000000000B30E0000640(	PORTO
8	01000001006172BF3E4D5A374080318D6907CA3EC0704734E198F3E5C082283B553DAA0441000000000000000000000000000000B30E0000640(	PORTO
9	01000001006172BF3E4D5A374080318D6907CA3EC05AEEA31BB0CFE4C082283B553DAA0441000000000000000000000000000000B30E000064	PORTO
10	01000001006172BF3E4D5A374080318D6907CA3EC0704734E198F3E5C0474F11FE054A0441000000000000000000000000000000B30E0000640(	PORTO
11	01000001006172BF3E4D5A374080318D6907CA3EC05AEEA31BB0CFE4C0474F11FE054A0441000000000000000000000000000000B30E0000640(	PORTO
12	01000001006172BF3E4D5A374080318D6907CA3EC085A0C4A68117E7C082283B553DAA0441000000000000000000000000000000B30E000064	PORTO
13	01000001006172BF3E4D5A374080318D6907CA3EC0704734E198F3E5C082283B553DAA0441000000000000000000000000000000B30E0000640(	PORTO
14	01000001006172BF3E4D5A374080318D6907CA3EC085A0C4A68117E7C0474F11FE054A0441000000000000000000000000000000B30E0000640	PORTO
15	01000001006172BF3E4D5A374080318D6907CA3EC0704734E198F3E5C0474F11FE054A0441000000000000000000000000000000B30E0000640(	PORTO
16	01000001006172BF3E4D5A374080318D6907CA3EC05AEEA31BB0CFE4C082283B553DAA0441000000000000000000000000000000B30E000064	PORTO
17	01000001006172BF3E4D5A374080318D6907CA3EC044951356C7ABE3C082283B553DAA0441000000000000000000000000000000B30E000064	PORTO
18	01000001006172BF3E4D5A374080318D6907CA3EC02E3C8390DE87E2C082283B553DAA0441000000000000000000000000000000B30E000064	PORTO
19	01000001006172BF3E4D5A374080318D6907CA3EC044951356C7ABE3C0474F11FE054A0441000000000000000000000000000000B30E0000640	PORTO

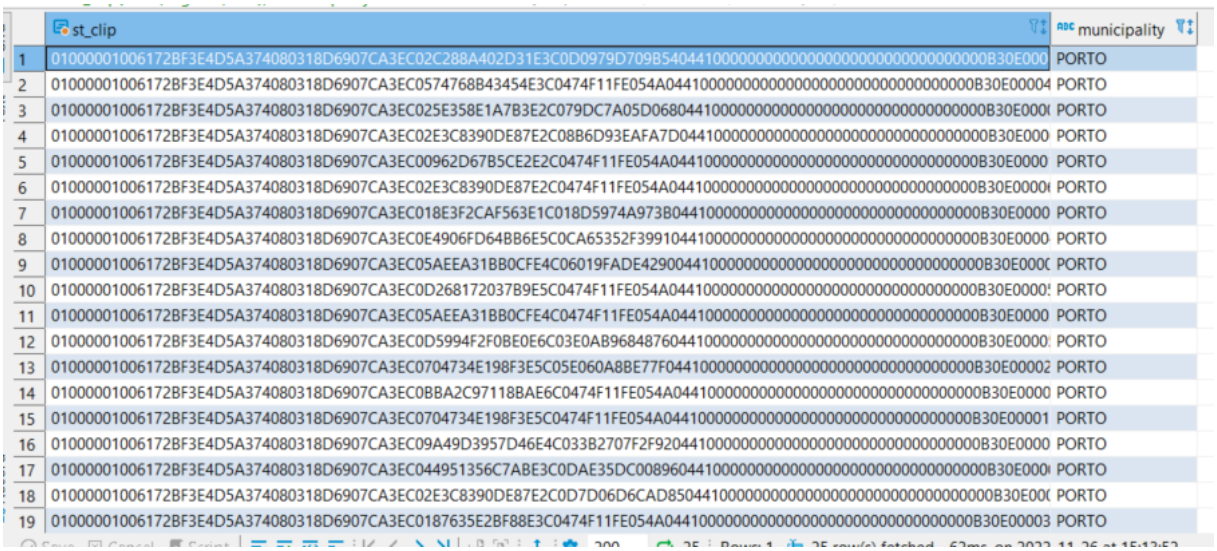


	rast	municipality	rid
1	01000001006172BF3E4D5A374080318D6907CA3EC044951356C7ABE3C082283B553DAA0441000000000000000000000000000000	PORTO	1
2	01000001006172BF3E4D5A374080318D6907CA3EC044951356C7ABE3C0474F11FE054A0441000000000000000000000000000000	PORTO	2
3	01000001006172BF3E4D5A374080318D6907CA3EC044951356C7ABE3C082283B553DAA0441000000000000000000000000000000	PORTO	3
4	01000001006172BF3E4D5A374080318D6907CA3EC02E3C8390DE87E2C082283B553DAA0441000000000000000000000000000000	PORTO	4
5	01000001006172BF3E4D5A374080318D6907CA3EC044951356C7ABE3C0474F11FE054A0441000000000000000000000000000000	PORTO	5
6	01000001006172BF3E4D5A374080318D6907CA3EC02E3C8390DE87E2C0474F11FE054A0441000000000000000000000000000000	PORTO	6
7	01000001006172BF3E4D5A374080318D6907CA3EC018E3F2CAF563E1C0474F11FE054A0441000000000000000000000000000000	PORTO	7
8	01000001006172BF3E4D5A374080318D6907CA3EC0704734E198F3E5C082283B553DAA0441000000000000000000000000000000	PORTO	8
9	01000001006172BF3E4D5A374080318D6907CA3EC05AEEA31BB0CFE4C082283B553DAA0441000000000000000000000000000000	PORTO	9
10	01000001006172BF3E4D5A374080318D6907CA3EC0704734E198F3E5C0474F11FE054A0441000000000000000000000000000000	PORTO	10
11	01000001006172BF3E4D5A374080318D6907CA3EC05AEEA31BB0CFE4C0474F11FE054A0441000000000000000000000000000000	PORTO	11
12	01000001006172BF3E4D5A374080318D6907CA3EC085A0C4A68117E7C082283B553DAA0441000000000000000000000000000000	PORTO	12
13	01000001006172BF3E4D5A374080318D6907CA3EC0704734E198F3E5C082283B553DAA0441000000000000000000000000000000	PORTO	13
14	01000001006172BF3E4D5A374080318D6907CA3EC085A0C4A68117E7C0474F11FE054A0441000000000000000000000000000000	PORTO	14
15	01000001006172BF3E4D5A374080318D6907CA3EC0704734E198F3E5C0474F11FE054A0441000000000000000000000000000000	PORTO	15
16	01000001006172BF3E4D5A374080318D6907CA3EC05AEEA31BB0CFE4C082283B553DAA0441000000000000000000000000000000	PORTO	16
17	01000001006172BF3E4D5A374080318D6907CA3EC044951356C7ABE3C082283B553DAA0441000000000000000000000000000000	PORTO	17
18	01000001006172BF3E4D5A374080318D6907CA3EC02E3C8390DE87E2C082283B553DAA0441000000000000000000000000000000	PORTO	18
19	01000001006172BF3E4D5A374080318D6907CA3EC044951356C7ABE3C0474F11FE054A0441000000000000000000000000000000	PORTO	19



```
Output
Adding SRID constraint
Adding scale-X constraint
Adding scale-Y constraint
Adding blocksize-X constraint
Adding blocksize-Y constraint
Adding alignment constraint
Adding number of bands constraint
Adding pixel type constraint
Adding nodata value constraint
Adding out-of-database constraint
Adding maximum extent constraint
```

-- 2 przykład



	st_clip	municipality
1	01000001006172BF3E4D5A374080318D6907CA3EC02C288A402D31E3C0D0979D709B5404410000000000000000000000000000000B30E0000	PORTO
2	01000001006172BF3E4D5A374080318D6907CA3EC0574768B43454E3C0474F11FE054A0441000000000000000000000000000000B30E00004	PORTO
3	01000001006172BF3E4D5A374080318D6907CA3EC025E358E1A7B3E2C079DC7A05D0680441000000000000000000000000000000B30E0000(	PORTO
4	01000001006172BF3E4D5A374080318D6907CA3EC02E3C8390DE87E2C08B6D93EAFA7D0441000000000000000000000000000000B30E0000	PORTO
5	01000001006172BF3E4D5A374080318D6907CA3EC00962D67B5CE2E2C0474F11FE054A0441000000000000000000000000000000B30E0000	PORTO
6	01000001006172BF3E4D5A374080318D6907CA3EC02E3C8390DE87E2C0474F11FE054A0441000000000000000000000000000000B30E0000(	PORTO
7	01000001006172BF3E4D5A374080318D6907CA3EC018E3F2CAF563E1C018D5974A973B0441000000000000000000000000000000B30E0000	PORTO
8	01000001006172BF3E4D5A374080318D6907CA3EC0E4906FD64BB6E5C0CA65352F3991044100000000000000000000000000000B30E0000	PORTO
9	01000001006172BF3E4D5A374080318D6907CA3EC05AEEA31BB0CFE4C06019FADE4290044100000000000000000000000000000B30E0000(	PORTO
10	01000001006172BF3E4D5A374080318D6907CA3EC0D268172037B9E5C0474F11FE054A0441000000000000000000000000000000B30E0000:	PORTO
11	01000001006172BF3E4D5A374080318D6907CA3EC05AEEA31BB0CFE4C0474F11FE054A0441000000000000000000000000000000B30E0000	PORTO
12	01000001006172BF3E4D5A374080318D6907CA3EC0D5994F2F0BE0E6C03E0AB96848760441000000000000000000000000000000B30E0000:	PORTO
13	01000001006172BF3E4D5A374080318D6907CA3EC0704734E198F3E5C05E060A8BE77F0441000000000000000000000000000000B30E00002	PORTO
14	01000001006172BF3E4D5A374080318D6907CA3EC0BBA2C97118BAE6C0474F11FE054A0441000000000000000000000000000000B30E0000	PORTO
15	01000001006172BF3E4D5A374080318D6907CA3EC0704734E198F3E5C0474F11FE054A0441000000000000000000000000000000B30E00001	PORTO
16	01000001006172BF3E4D5A374080318D6907CA3EC09A49D3957D46E4C033B2707F2F9204410000000000000000000000000000000B30E0000	PORTO
17	01000001006172BF3E4D5A374080318D6907CA3EC044951356C7ABE3C0DAE35DC0089604410000000000000000000000000000000B30E0000(	PORTO
18	01000001006172BF3E4D5A374080318D6907CA3EC02E3C8390DE87E2C0D7D06D6CAD850441000000000000000000000000000000B30E000(	PORTO
19	01000001006172BF3E4D5A374080318D6907CA3EC0187635E2BF88E3C0474F11FE054A0441000000000000000000000000000000B30E00003	PORTO

-- 3 przykład

	st_union
1	01000001006172BF3E4D5A374080318D6907CA3EC0D6994F2F0BE0E6C0D9E35DC0089604410000000000000000000000000000000B30E0000F201AC0045018001

Tworzenie rastrów z wektorów (rastrowanie)

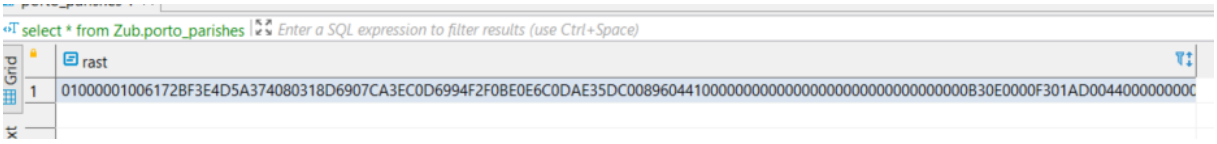
-- przykład 1

select * from Zub.porto_par | Enter a SQL expression to filter results (use Ctrl+Space)

Grid | Text

	rast
1	01000001006172BF3E4D5A374080318D6907CA3EC0574768B43454E3C0D(
2	01000001006172BF3E4D5A374080318D6907CA3EC00A62D67B5CE2E2C08
3	01000001006172BF3E4D5A374080318D6907CA3EC0D268172037B9E5C0C/
4	01000001006172BF3E4D5A374080318D6907CA3EC0D6994F2F0BE0E6C05E
5	01000001006172BF3E4D5A374080318D6907CA3EC09A49D3957D46E4C0D
6	01000001006172BF3E4D5A374080318D6907CA3EC087977E37109EE4C0A3
7	01000001006172BF3E4D5A374080318D6907CA3EC00FB04D4A53D9E5C04

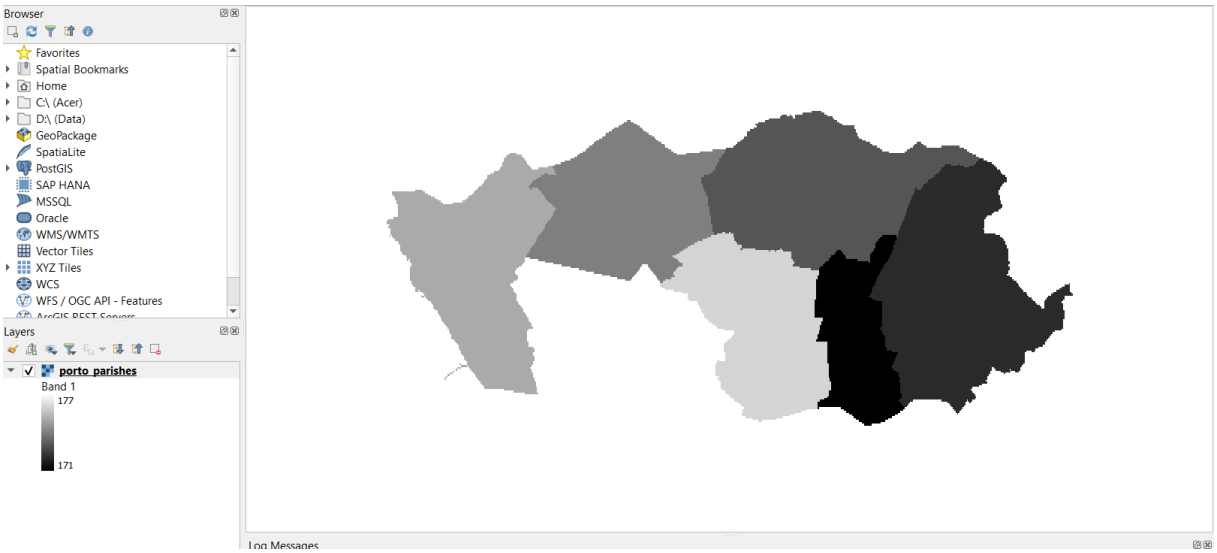
-- przykład 2



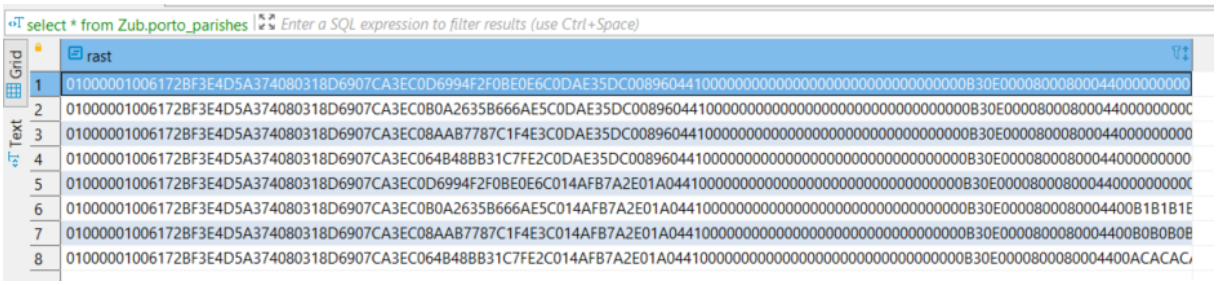
select * from Zub.porto_parishes | Enter a SQL expression to filter results (use Ctrl+Space)

Grid

	rast
1	01000001006172BF3E4D5A374080318D6907CA3EC0D6994F2F0BE0E6C0DAE35DC0089604410000000000000000000000000000000B30E0000F301AD00440000000(

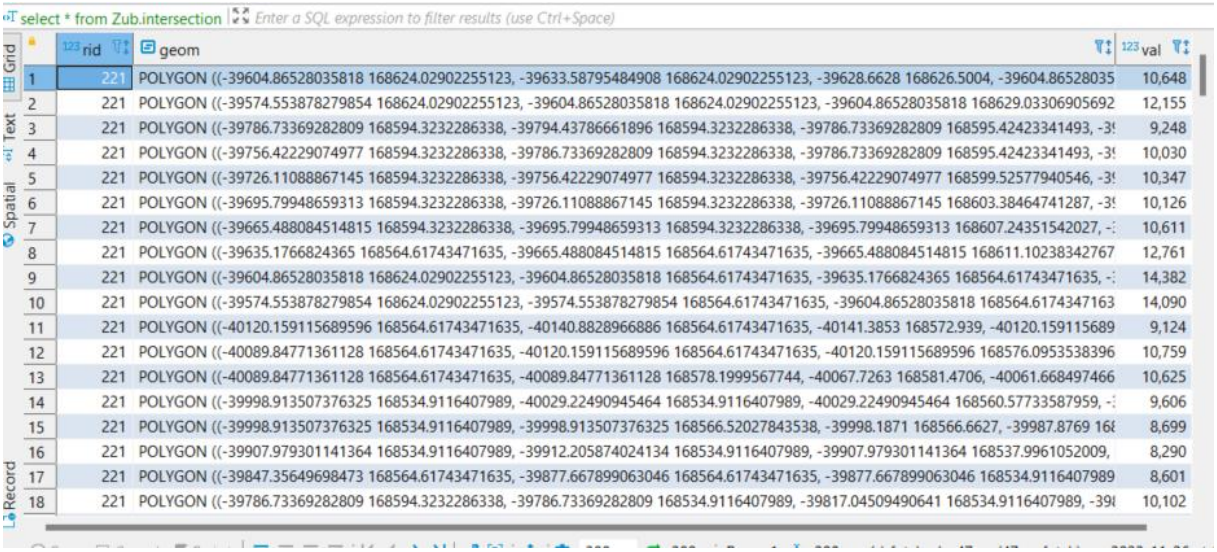


-- przykład 3

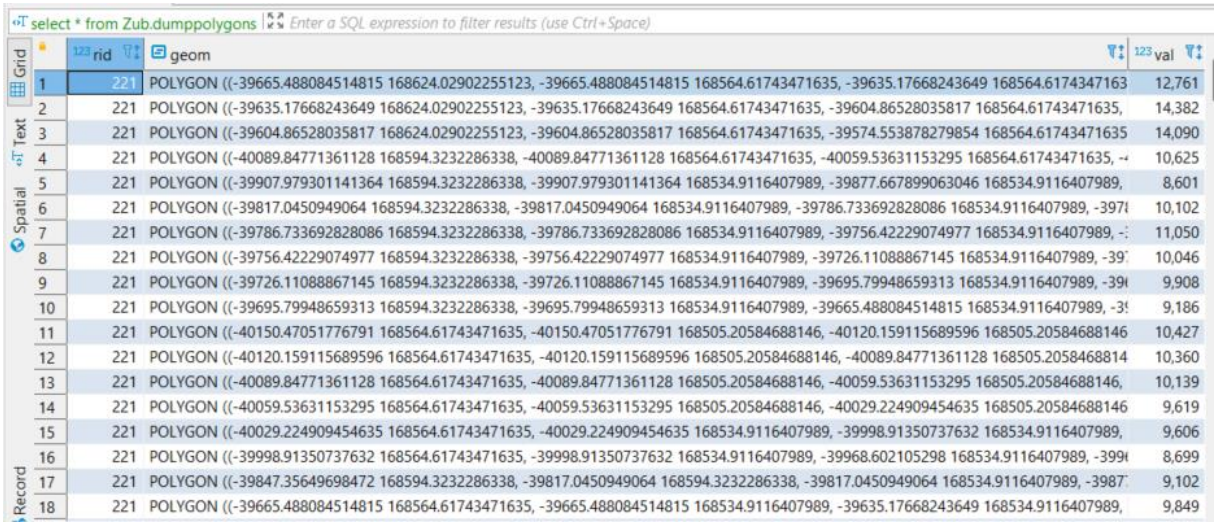


Konwertowanie rastrów na wektory (wektoryzowanie)

-- przykład 1



-- przykład 2

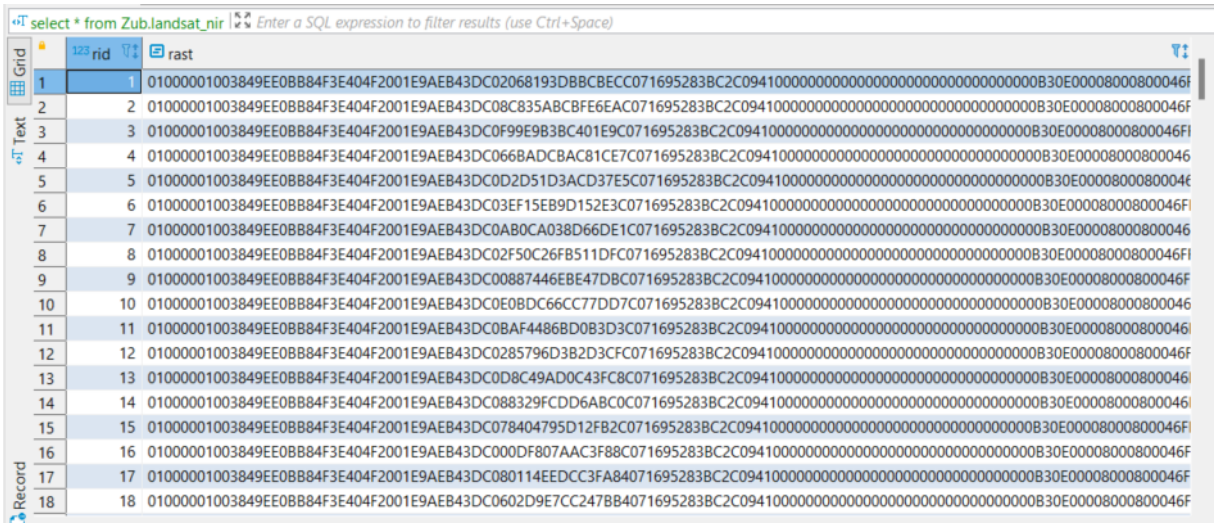


select * from Zub.dumppolygons | Enter a SQL expression to filter results (use Ctrl+Space)

	rid	geom	val
1	221	POLYGON ((-39665.488084514815 168624.02902255123, -39665.488084514815 168564.61743471635, -39635.17668243649 168564.6174347163	12,761
2	221	POLYGON ((-39635.17668243649 168624.02902255123, -39635.17668243649 168564.61743471635, -39604.86528035817 168564.61743471635,	14,382
3	221	POLYGON ((-39604.86528035817 168624.02902255123, -39604.86528035817 168564.61743471635, -39574.553878279854 168564.61743471635	14,090
4	221	POLYGON ((-40089.84771361128 168594.3232286338, -40089.84771361128 168564.61743471635, -40059.53631153295 168564.61743471635, -4	10,625
5	221	POLYGON ((-39907.979301141364 168594.3232286338, -39907.979301141364 168534.9116407989, -39877.667899063046 168534.9116407989,	8,601
6	221	POLYGON ((-39817.0450949064 168594.3232286338, -39817.0450949064 168534.9116407989, -39786.733692828086 168534.9116407989, -397	10,102
7	221	POLYGON ((-39786.733692828086 168594.3232286338, -39786.733692828086 168534.9116407989, -39756.42229074977 168534.9116407989, -3	11,050
8	221	POLYGON ((-39756.42229074977 168594.3232286338, -39756.42229074977 168534.9116407989, -39726.11088867145 168534.9116407989, -39	10,046
9	221	POLYGON ((-39726.11088867145 168594.3232286338, -39726.11088867145 168534.9116407989, -39695.79948659313 168534.9116407989, -39	9,908
10	221	POLYGON ((-39695.79948659313 168594.3232286338, -39695.79948659313 168534.9116407989, -39665.488084514815 168534.9116407989, -3	9,186
11	221	POLYGON ((-40150.47051776791 168564.61743471635, -40150.47051776791 168505.20584688146, -40120.159115689596 168505.20584688146	10,427
12	221	POLYGON ((-40120.159115689596 168564.61743471635, -40120.159115689596 168505.20584688146, -40089.84771361128 168505.2058468814	10,360
13	221	POLYGON ((-40089.84771361128 168564.61743471635, -40089.84771361128 168505.20584688146, -40059.53631153295 168505.20584688146,	10,139
14	221	POLYGON ((-40059.53631153295 168564.61743471635, -40059.53631153295 168505.20584688146, -40029.224909454635 168505.20584688146	9,619
15	221	POLYGON ((-40029.224909454635 168564.61743471635, -40029.224909454635 168534.9116407989, -39998.91350737632 168534.9116407989,	9,606
16	221	POLYGON ((-39998.91350737632 168564.61743471635, -39998.91350737632 168534.9116407989, -39968.602105298 168534.9116407989, -399	8,699
17	221	POLYGON ((-39847.35649698472 168594.3232286338, -39817.0450949064 168594.3232286338, -39817.0450949064 168534.9116407989, -3987	9,102
18	221	POLYGON ((-39665.488084514815 168564.61743471635, -39665.488084514815 168534.9116407989, -39635.17668243649 168534.9116407989,	9,849

Analiza rastrów

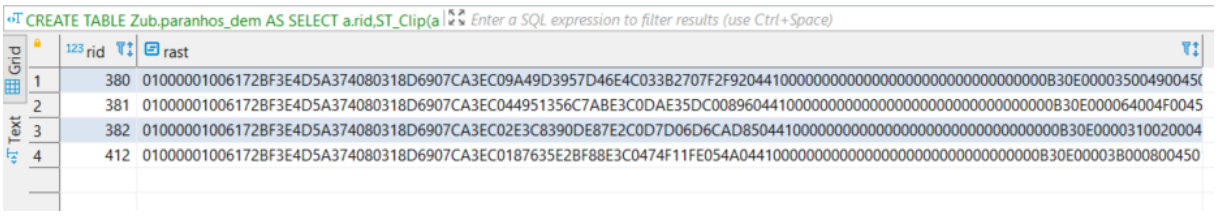
-- przykład 1



select * from Zub.landsat_nir | Enter a SQL expression to filter results (use Ctrl+Space)

	rid	rast
1	1	0100000100384 9EE0BB84F3E404F2001E9AEB43DC02068193DBBCBECC071695283BC2C0941000000000000000000000000000000000B30E000080008004 6
2	2	0100000100384 9EE0BB84F3E404F2001E9AEB43DC08C835ABCBFE6EAC071695283BC2C0941000000000000000000000000000000000B30E000080008004 6
3	3	0100000100384 9EE0BB84F3E404F2001E9AEB43DC0F99E9B3BC401E9C071695283BC2C0941000000000000000000000000000000000B30E000080008004 6
4	4	0100000100384 9EE0BB84F3E404F2001E9AEB43DC066BADCBAC81CE7C071695283BC2C0941000000000000000000000000000000000B30E000080008004 6
5	5	0100000100384 9EE0BB84F3E404F2001E9AEB43DC0D2D51D3ACD37E5C071695283BC2C0941000000000000000000000000000000000B30E000080008004 6
6	6	0100000100384 9EE0BB84F3E404F2001E9AEB43DC03EF15EB9D152E3C071695283BC2C0941000000000000000000000000000000000B30E000080008004 6
7	7	0100000100384 9EE0BB84F3E404F2001E9AEB43DC0AB0CA038D66DE1C071695283BC2C0941000000000000000000000000000000000B30E000080008004 6
8	8	0100000100384 9EE0BB84F3E404F2001E9AEB43DC02F50C26FB511DFC071695283BC2C0941000000000000000000000000000000000B30E000080008004 6
9	9	0100000100384 9EE0BB84F3E404F2001E9AEB43DC00887446EBE47DBC071695283BC2C0941000000000000000000000000000000000B30E000080008004 6
10	10	0100000100384 9EE0BB84F3E404F2001E9AEB43DC0E0BDC66CC77DD7C071695283BC2C0941000000000000000000000000000000000B30E000080008004 6
11	11	0100000100384 9EE0BB84F3E404F2001E9AEB43DC0BAF4486BD0B3D3C071695283BC2C0941000000000000000000000000000000000B30E000080008004 6
12	12	0100000100384 9EE0BB84F3E404F2001E9AEB43DC0285796D3B2D3CFC071695283BC2C0941000000000000000000000000000000000B30E000080008004 6
13	13	0100000100384 9EE0BB84F3E404F2001E9AEB43DC0D8C49AD0C43FC8C071695283BC2C0941000000000000000000000000000000000B30E000080008004 6
14	14	0100000100384 9EE0BB84F3E404F2001E9AEB43DC088329FCDD6ABC0C071695283BC2C0941000000000000000000000000000000000B30E000080008004 6
15	15	0100000100384 9EE0BB84F3E404F2001E9AEB43DC078404795D12FB2C071695283BC2C0941000000000000000000000000000000000B30E000080008004 6
16	16	0100000100384 9EE0BB84F3E404F2001E9AEB43DC000DF807AAC3F88C071695283BC2C0941000000000000000000000000000000000B30E000080008004 6
17	17	0100000100384 9EE0BB84F3E404F2001E9AEB43DC080114EEDCC3FA84071695283BC2C0941000000000000000000000000000000000B30E000080008004 6
18	18	0100000100384 9EE0BB84F3E404F2001E9AEB43DC0602D9E7CC247BB4071695283BC2C0941000000000000000000000000000000000B30E000080008004 6

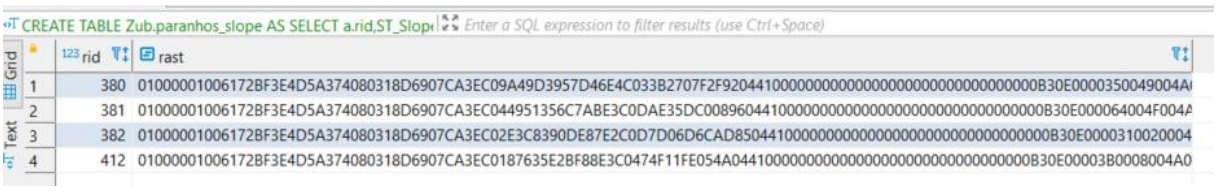
--przykład 2



CREATE TABLE Zub.paranhos_dem AS SELECT a.rid,ST_Clip(a | Enter a SQL expression to filter results (use Ctrl+Space)

	rid	rast
1	380	01000001006172BF3E4D5A374080318D6907CA3EC09A49D3957D46E4C033B2707F2F92044100000000000000000000000000000000B30E00003500490045(
2	381	01000001006172BF3E4D5A374080318D6907CA3EC044951356C7ABE3C0DAE35DC00896044100000000000000000000000000000000B30E000064004F0045
3	382	01000001006172BF3E4D5A374080318D6907CA3EC02E3C8390DE87E2C0D7D06D6CAD85044100000000000000000000000000000000B30E0000310020004
4	412	01000001006172BF3E4D5A374080318D6907CA3EC0187635E2BF88E3C0474F11FE054A04410000000000000000000000000000000B30E00003B000800450

-- przykład 3



CREATE TABLE Zub.paranhos_slope AS SELECT a.rid,ST_Slope | Enter a SQL expression to filter results (use Ctrl+Space)

	rid	rast
1	380	01000001006172BF3E4D5A374080318D6907CA3EC09A49D3957D46E4C033B2707F2F92044100000000000000000000000000000000B30E00003500490044
2	381	01000001006172BF3E4D5A374080318D6907CA3EC044951356C7ABE3C0DAE35DC00896044100000000000000000000000000000000B30E000064004F004A
3	382	01000001006172BF3E4D5A374080318D6907CA3EC02E3C8390DE87E2C0D7D06D6CAD85044100000000000000000000000000000000B30E0000310020004
4	412	01000001006172BF3E4D5A374080318D6907CA3EC0187635E2BF88E3C0474F11FE054A04410000000000000000000000000000000B30E00003B0008004A0

-- przykład 4

paranhos_slope_reclass 1

```
CREATE TABLE Zub.paranhos_slope_reclass AS SELECT a.rid,s
```

Enter a SQL expression to filter results (use Ctrl+Space)

	rid	st_reclass
1	380	01000001006172BF3E4D5A374080318D6907CA3EC09A49D3957D46E4C033B2707F2F92044100000000000000000000000000000000B30E0000350049004A
2	381	01000001006172BF3E4D5A374080318D6907CA3EC044951356C7ABE3C0DAE35DC0089604410000000000000000000000000000000B30E000064004F004A
3	382	01000001006172BF3E4D5A374080318D6907CA3EC02E3C8390DE87E2C0D7D06D6CAD850441000000000000000000000000000000B30E0000310020004
4	412	01000001006172BF3E4D5A374080318D6907CA3EC0187635E2BF88E3C0474F11FE054A0441000000000000000000000000000000B30E00003B0008004A0

-- przykład 5

```
SELECT st_summarystats(a.rast) AS stats FROM Zub.paranhos
```

Enter a SQL expression to filter results (use C

	stats					
	count	sum	mean	stddev	min	max
1	2,616	278,385	106.4162844037	11.6226287622	87	143
2	6,463	816,615	126.3523131673	14.0438229209	94	158
3	682	95,581	140.1480938416	12.0780721866	103	158
4	216	31,874	147.5648148148	4.2628306283	137	158

-- przykład 6

```
SELECT st_summarystats(ST_Union(a.rast)) FROM Zub.paranh
```

Enter a SQL expression to filter results (use

	st_summarystats					
	count	sum	mean	stddev	min	max
1	9,977	1,222,455	122.5273128195	16.9080042027	87	158

--przykład 7

```
WITH t AS ( SELECT st_summarystats(ST_Union(a.rast)) AS st
```

Enter a SQL ex

	min	max	mean
1	87	158	122.5273128195

-- przykład 8

```
WITH t AS ( SELECT b.parish AS parish, st_summarystats(ST_U
```

Enter a SQL expression to filter results (use Ctrl+Space)

	parish	min	max	mean
1	Bonfim	1	159	107.5658842668
2	Campanhã	0	178	74.6673221309
3	Paranhos	87	158	122.5273128195
4	Ramalde	48	108	77.5844444444
5	União das freguesias de Aldoar, Foz do Douro e Nevogilde	-4	83	34.6673548979
6	União das freguesias de Cedofeita, Santo Ildefonso, Sé, Miragaia, São Nicolau e Vitória	1	157	95.0027774104
7	União das freguesias de Lordelo do Ouro e Massarelos	-1	117	49.5005144033

-- przykład 9

```
SELECT b.name,st_value(a.rast,(ST_Dump(b.geom)).geom)
```

	name	st_value
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
15	Guimarães	197

-- przykład 10

32.6s

tpi30 1 | Output

```
select * from Zub.tpi30
```

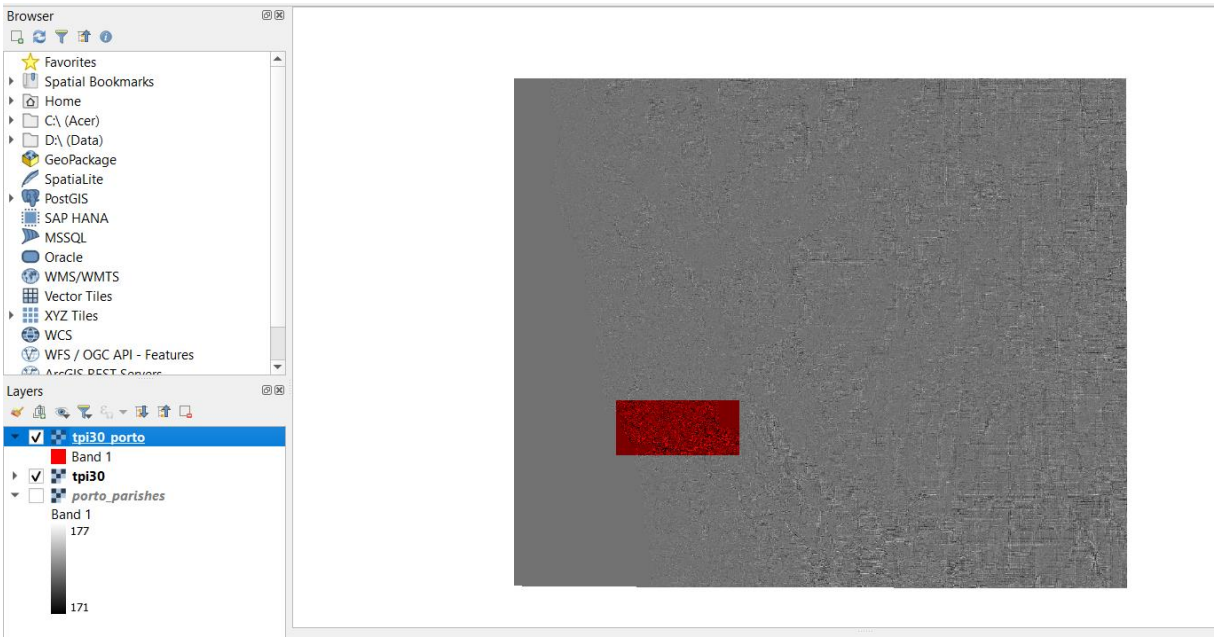
Enter a SQL expression to filter results (use Ctrl+Space)

	rast
1	01000001006172BF3E4D5A374080318D6907CA3EC0F25D96820DCBECC04257316BD52C0941000000000000000000000000
2	01000001006172BF3E4D5A374080318D6907CA3EC0DC0406BD24A7EBC04257316BD52C0941000000000000000000000000
3	01000001006172BF3E4D5A374080318D6907CA3EC0C6AB75F73B83EAC04257316BD52C0941000000000000000000000000
4	01000001006172BF3E4D5A374080318D6907CA3EC0B152E531535FE9C04257316BD52C0941000000000000000000000000
5	01000001006172BF3E4D5A374080318D6907CA3EC09BF9546C6A3BE8C04257316BD52C0941000000000000000000000000
6	01000001006172BF3E4D5A374080318D6907CA3EC085A0C4A68117E7C04257316BD52C0941000000000000000000000000
7	01000001006172BF3E4D5A374080318D6907CA3EC0704734E198F3E5C04257316BD52C0941000000000000000000000000
8	01000001006172BF3E4D5A374080318D6907CA3EC05AEEA31BB0CFE4C04257316BD52C0941000000000000000000000000
9	01000001006172BF3E4D5A374080318D6907CA3EC044951356C7ABE3C04257316BD52C0941000000000000000000000000
10	01000001006172BF3E4D5A374080318D6907CA3EC02E3C8390DE87E2C04257316BD52C0941000000000000000000000000
11	01000001006172BF3E4D5A374080318D6907CA3EC018E3F2CAF563E1C04257316BD52C0941000000000000000000000000

-- Problem do samodzielnego rozwiązania

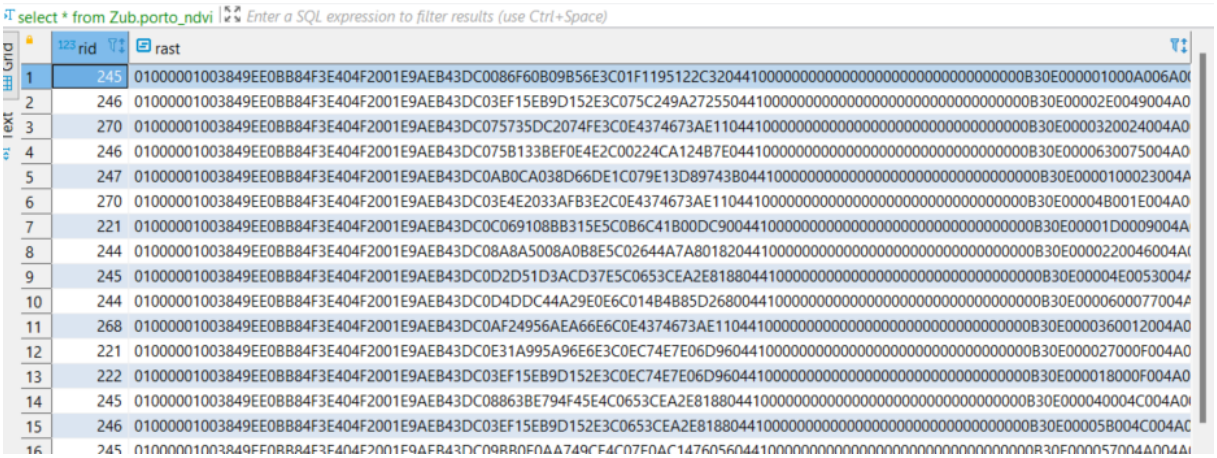
1.7s

```sql
--problem
create table Zub.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';
```

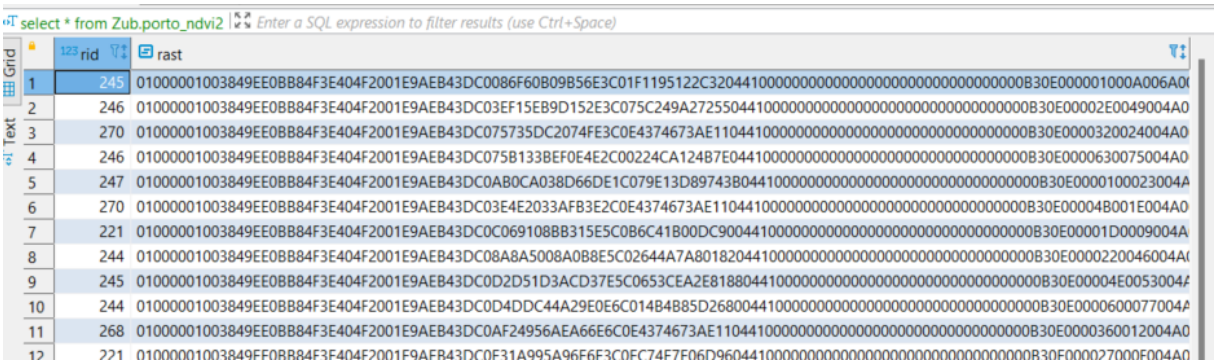


Algebra map

-- przykład 1



-- przykład 2



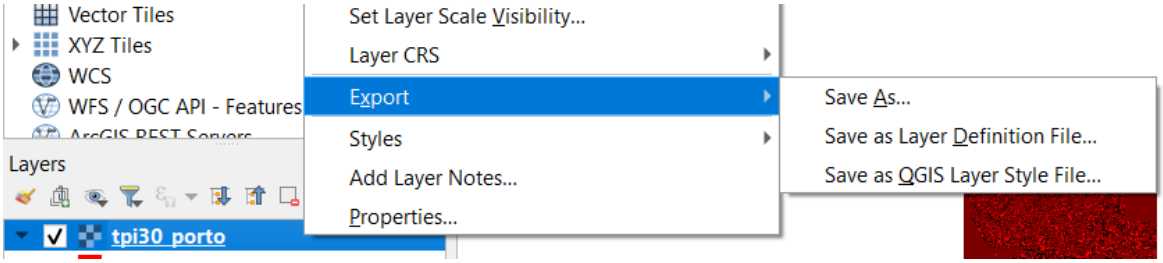
-- przykład 3

```
CREATE OR REPLACE FUNCTION public._st_tpi4ma(value double precision[], pos integer[], VARIADIC userargs text[] DEFAULT NULL::text[])
 RETURNS double precision
 LANGUAGE plpgsql
 IMMUTABLE PARALLEL SAFE
AS $function$
    DECLARE
        x integer;
        y integer;
        z integer;

        Z1 double precision;
        Z2 double precision;
        Z3 double precision;
        Z4 double precision;
        Z5 double precision;
        Z6 double precision;
        Z7 double precision;
        Z8 double precision;
        Z9 double precision;

        tpi double precision;
        mean double precision;
        _value double precision[][][];
        ndims int;
    BEGIN
        ndims := array_ndims(value);
        -- add a third dimension if 2-dimension
        IF ndims = 2 THEN
            _value := public._ST_convertarray4ma(value);
        ELSEIF ndims != 3 THEN
            RAISE EXCEPTION 'First parameter of function must be a 3-dimension array';
        ELSE
            _value := value;
        END IF;

        -- only use the first raster passed to this function
        IF array_length(_value, 1) > 1 THEN
            RAISE NOTICE 'Only using the values from the first raster';
        END IF;
        z := array_lower(_value, 1);

        IF (
            array_lower(_value, 2) != 1 OR array_upper(_value, 2) != 3 OR
            array_lower(_value, 3) != 1 OR array_upper(_value, 3) != 3
        ) THEN
            RAISE EXCEPTION 'First parameter of function must be a 1x3x3 array with each of the lower bounds starting from 1';
        END IF;

        -- check that center pixel isn't NODATA
        IF _value[z][2][2] IS NULL THEN
            RETURN NULL;
        -- substitute center pixel for any neighbor pixels that are NODATA
        ELSE
```

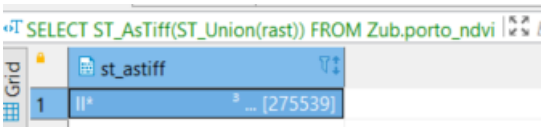
```
CREATE OR REPLACE FUNCTION public.st_tpi(rast raster, nband integer DEFAULT 1, pixeltype text DEFAULT '32BF'::text, interpolate_nodata boolean DEFAULT false)
 RETURNS raster
 LANGUAGE sql
 IMMUTABLE PARALLEL SAFE
AS $function$ SELECT public.ST_tpi($1, $2, NULL::public.raster, $3, $4) $function$
;
```

Eksport danych

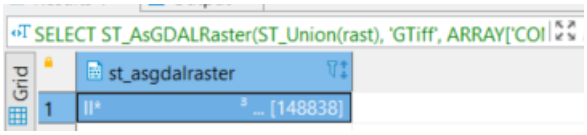
-- przykład 0

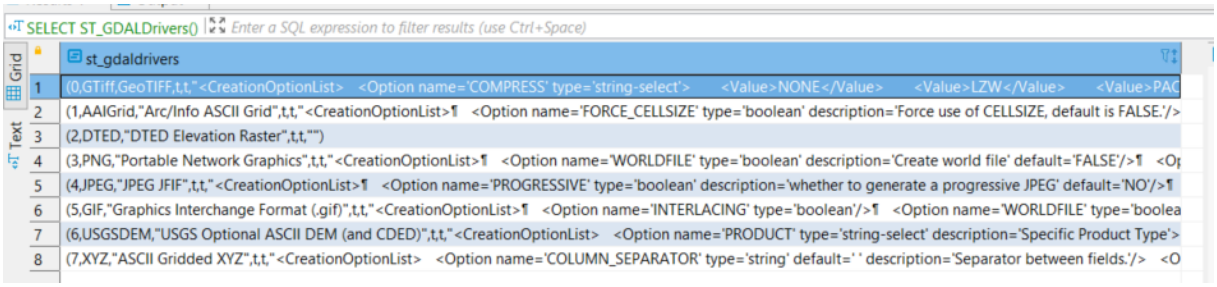


-- przykład 1

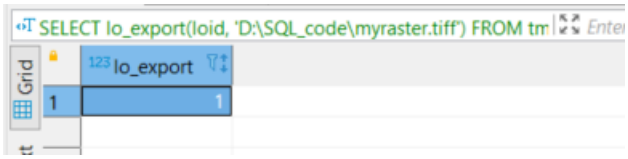


-- przykład 2



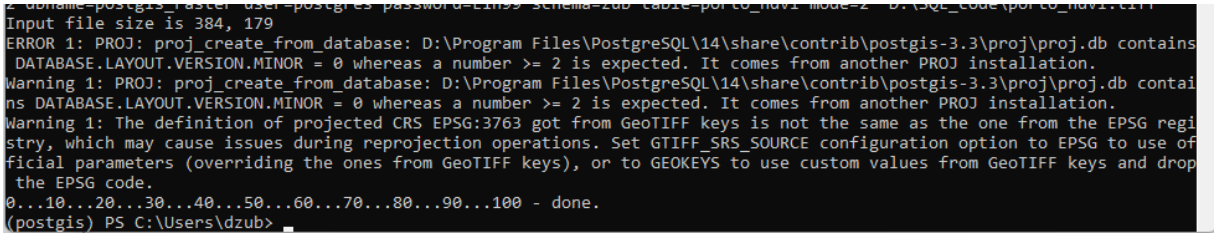


-- przykład 3

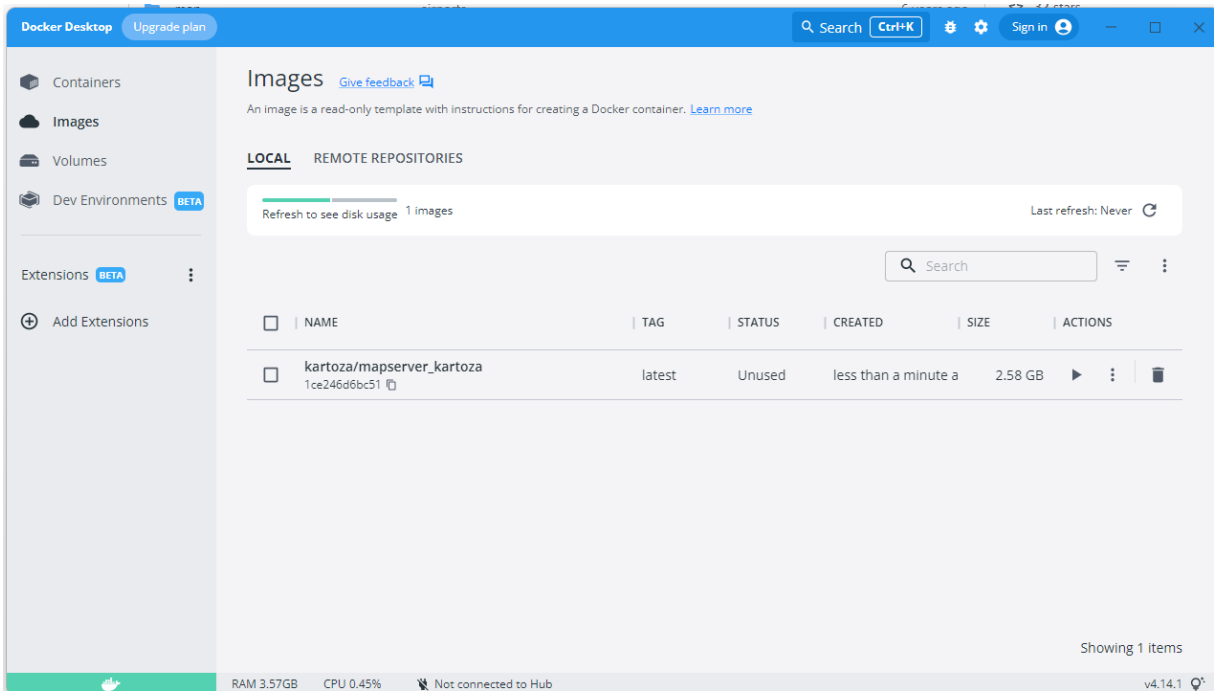


-- przykład 4

gdal_translate -co COMPRESS=DEFLATE -co PREDICTOR=2 -co ZLEVEL=9 PG:"host=localhost port=5432 dbname=postgis_raster user=postgres password=postgis schema=zub table=porto_ndvi mode=2" D:\SQL_code\porto_ndvi.tiff

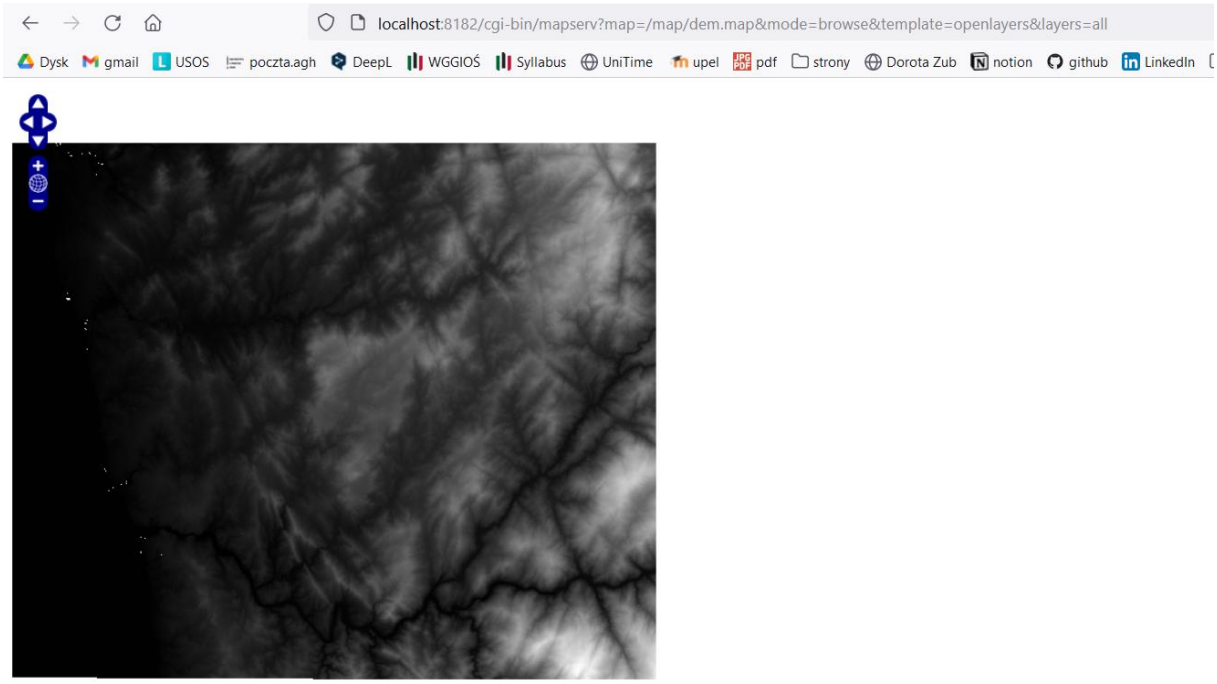


Publikowanie danych za pomocą MapServer




```
dzub@LAPTOP-AQDA90H2:~/bdp/docker-mapserver$ docker run -d -p 8182:80 --name mapserver_cont kartoza/mapserver_kartoza
9651950092a12fe41bf427462d645cefe2f9b80e4a90cdb1b83ce90b9d065626
dzub@LAPTOP-AQDA90H2:~/bdp/docker-mapserver$ docker exec -it mapserver_cont /bin/bash
root@9651950092a1:/# ls
bin   dev  home  lib32  libx32  mnt  proc  run   setup.sh  sys  usr
boot  etc  lib   lib64  media   opt  root  sbin  srv       tmp  var
root@9651950092a1:/# cd home
root@9651950092a1:/home# ls
```

```
psql (12.12 (Ubuntu 12.12-0ubuntu0.20.04.1), server 14.5)
WARNING: psql major version 12, server major version 14.
         Some psql features might not work.
Type "help" for help.

postgis_raster=# \dn
  List of schemas
  Name   |  Owner
---------+----------
 public  | postgres
 rasters | postgres
 vectors | postgres
 zub     | postgres
(4 rows)
```


localhost:8182/cgi-bin/mapserv?map=/map/dem.map&mode=browse&template=openlayers&layers=all
Dysk
gmail
USOS
poczta.agh
DeepL
WGGIOŚ
Syllabus
UniTime
upel
pdf
strony
Dorota Zub
notion
github
LinkedIn