

2.

```
raster2pgsql -e "D:\Moja edukacja\AGH\Semestr 7\Bazy danych  
przestrzennych\lab7\ras250_gb\data\*.tif" rasters.uk_250k | psql -d lab8 -h localhost -U postgres -p  
5432
```

3.

Po wczytaniu danych do bazy spróbowałem wykonać ST_UNION na wszystkich rastrach, niestety nie udało się to z powodu braku pamięci

```
SQL SELECT ST_Union(rast) from rasters.uk_250k uk limit 1: Enter a SQL expression to filter results (use Ctrl+Space)
```



SQL Error [XX000]: ERROR: invalid memory alloc request size 1152000000
Error position:

Details >>



Komenda wyglądała następująco:

```
SELECT ST_Union(rast)  
from rasters.uk_250k uk
```

Testowałem również opcje z funkcją LIMIT, jednak nawet dla LIMIT 5 błąd z pamięcią występował.

5.

W związku z problemami z wczytaniem danych, postanowiłem najpierw zapisać je do tymczasowego pliku shp a potem wczytywać standardowo:

```
ogr2ogr -f "ESRI Shapefile" "D:\Moja edukacja\AGH\Semestr 7\Bazy danych  
przestrzennych\lab7\temp.shp" "D:\Moja edukacja\AGH\Semestr 7\Bazy danych  
przestrzennych\lab7\OS_Open_Zoomstack\OS_Open_Zoomstack.gpkg"
```

Wczytanie pliku shp:

```
shp2pgsql -e "D:\Moja edukacja\AGH\Semestr 7\Bazy danych przestrzennych\lab7\temp.shp  
\national_parks.shp" public.uk_lake_district | psql -d lab8 -h localhost -U postgres -p 5432
```

6.

```
create table uk_lake_district as  
  select st_union(st_clip(uk.rast, n.geom, true))  
  from uk_250k uk  
  inner join national_parks as n on st_intersects(n.geom, uk.rast)  
  where n.gid=1
```

7.

```
create table out_clip as
select lo_from_bytea(0, ST_AsGDALRaster(ST_Union(u.rast), 'GTiff',
array['COMPRESS=DEFLATE', 'PREDICTOR=2', 'PZLEVEL=9'])) as lo
from uk_lake_district u;
```

Niestety wystąpił błąd z typem GDAL z którym nie mogłem sobie poradzić. Jakby powyższa komenda zadziałała to zapisałbym obraz w formacie TIFF takim kodem:

```
select lo_export(lo, 'out7.tiff')
from out_clip;

select lo_unlink(lo)
from out_clip;
```

9.

```
Processing 1/14: D:\Moja edukacja\AGH\Semestr 7\Bazy danych przestrzennych\lab7\S2B_MSIL1C_20221130T112329_N0400_R037_T30UVF_202
21130T120448.SAFE\GRANULE\L1C_T30UVF_A029950_20221130T112331\IMG_DATA\T30UVF_20221130T112329_B01.jp2
Password for user postgres:
CREATE TABLE
Processing 2/14: D:\Moja edukacja\AGH\Semestr 7\Bazy danych przestrzennych\lab7\S2B_MSIL1C_20221130T112329_N0400_R037_T30UVF_202
21130T120448.SAFE\GRANULE\L1C_T30UVF_A029950_20221130T112331\IMG_DATA\T30UVF_20221130T112329_B02.jp2
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
WARNING: Raster with different alignment found in the set of rasters being converted to PostGIS raster
```

10.

```
create table ndvi as
WITH temp AS (
SELECT s.rid, ST_Clip(s.rast, st_transform(u.geom, 32630)) as rast
FROM sentinel s
INNER JOIN uk_lake_district u ON st_intersects(st_transform(u.geom, 32630), s.rast)
WHERE u.gid = 1
)
SELECT temp.rid,
ST_MapAlgebra(
temp.rast, 1,
temp.rast, 4,
'((([rast2.val] - [rast1.val]) / ([rast2.val])) + [rast1.val])'::float,
'32BF'::text
) as rast
from temp;
```

11.

Podobnie jak wcześniej przez problem z formatem gdal poniższa komenda nie zadziała, jednak kod wyglądałby tak jak poniżej:

```
--11
create table out_ndvi as
  select lo_from_bytea(0, ST_AsGDALRaster(ST_Union(n.rast), 'GTiff',
    array['COMPRESS=DEFLATE', 'PREDICTOR=2', 'PZLEVEL=9'])) as lo
  from ndvi n;

select lo_export(lo, 'out11.tiff')
from out_ndvi;

select lo_unlink(lo)
from out_ndvi;
```



SQL Error [XX000]: ERROR: invalid memory alloc request size 1152000000
Error position: