

2.

raster2pgsql.exe -s 3763 -N -32767 -t 100x100 -l -C -M -d

D:/Semestr_7/BazyDanychPrzestrzennych/lab7/ras250_gb/ras250_gb/data/*.tif uk_250k | psql -d lab7 -h localhost -U postgres -p 5432

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

COMMIT
VACUUM

C:\Program Files\PostgreSQL\15\bin>
```

3.

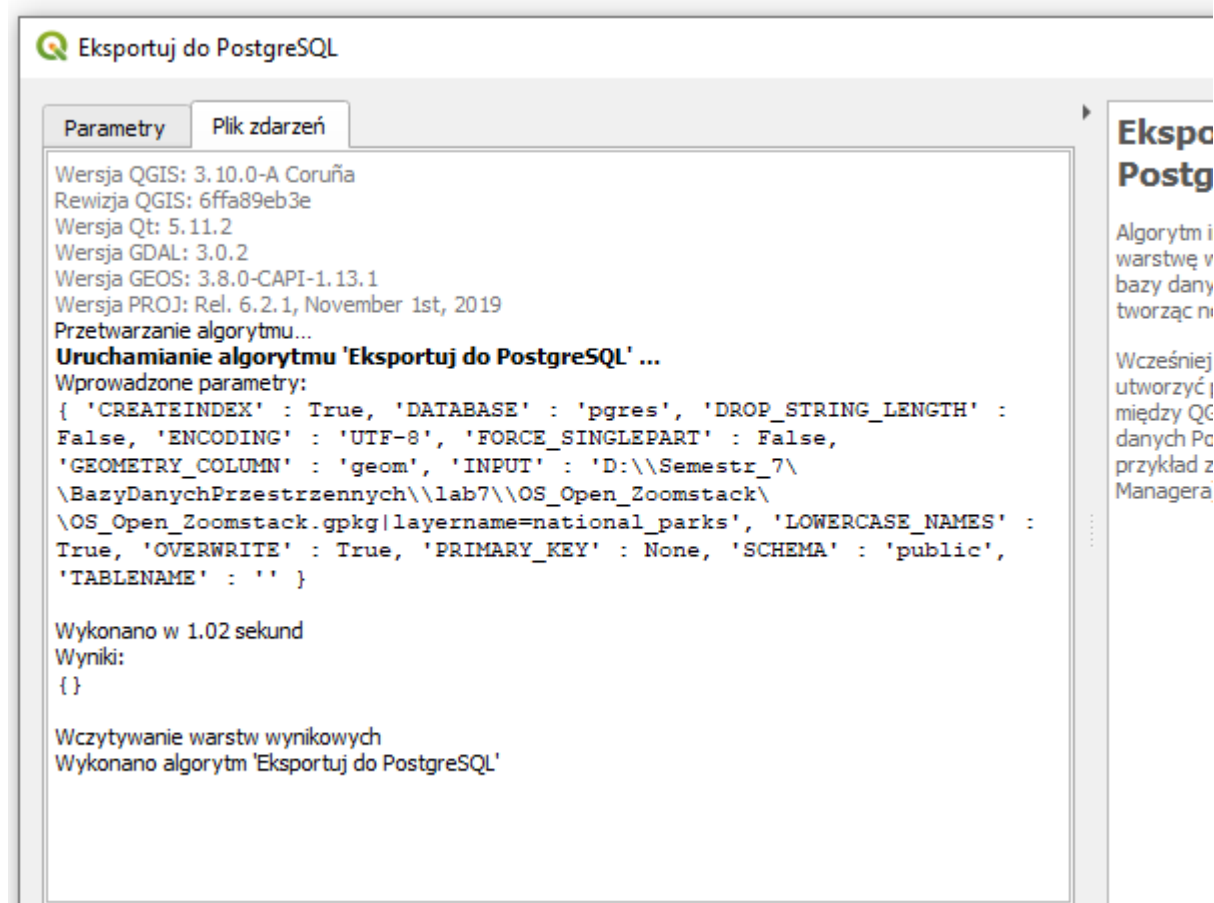
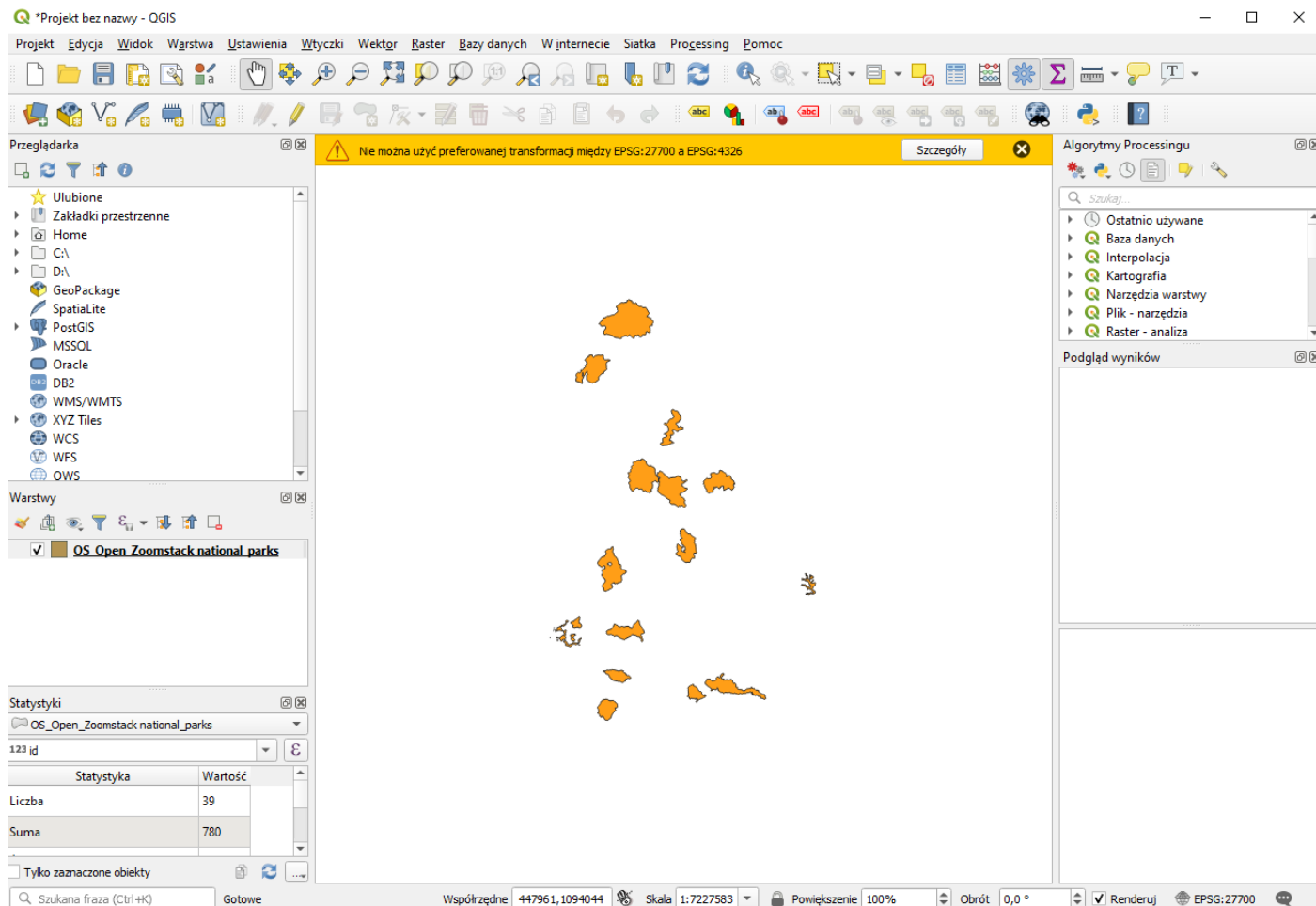
The screenshot shows the pgAdmin 4 web interface. The top navigation bar includes 'File', 'Object', 'Tools', and 'Help'. The 'Browser' tab is active, showing a tree of database objects. Under 'Servers (2)', 'PostgreSQL 15' is expanded, and 'Databases (6)' is selected. The 'lab7' database is highlighted. The main pane shows a SQL query editor with the following query:

```
1 SELECT ST_AsGDALRaster(ST_Union(rast), 'GTiff')
2 FROM uk_250k
```

The 'Data Output' tab at the bottom is empty, displaying a message: 'No data output. Execute a query to get output.' and 'Waiting for the query to complete...'.

Kręci się tak już prawie godzinę. Niestety to polecenie nie jest w stanie się wykonać, prawdopodobnie ze względu na brak pamięci

5.



Query
Query History

```

1 SELECT * FROM public."OS_Open_Zoomstacknational_parks"
2 ORDER BY id ASC LIMIT 100
3

```

Data Output
Messages
Notifications

+

📄

▼

📋

🗑️

🗄️

⬇️

📈

	id [PK] bigint	geom geometry
1	1	0103000020346...
2	2	0103000020346...
3	3	0103000020346...
4	4	0103000020346...
5	5	0103000020346...

6.

Query
Query History

```

1 CREATE TABLE uk_lake_disctrict AS
2 SELECT ST_Union(ST_Clip(A.rast, B.geom, true))
3 FROM public.uk_250k AS A, public."OS_Open_Zoomstacknational_parks" AS B
4 WHERE B.id = 1 and ST_Intersects(B.geom, A.rast);

```

Data Output
Messages
Notifications

SELECT 1

Query returned successfully in 7 secs 278 msec.

7.

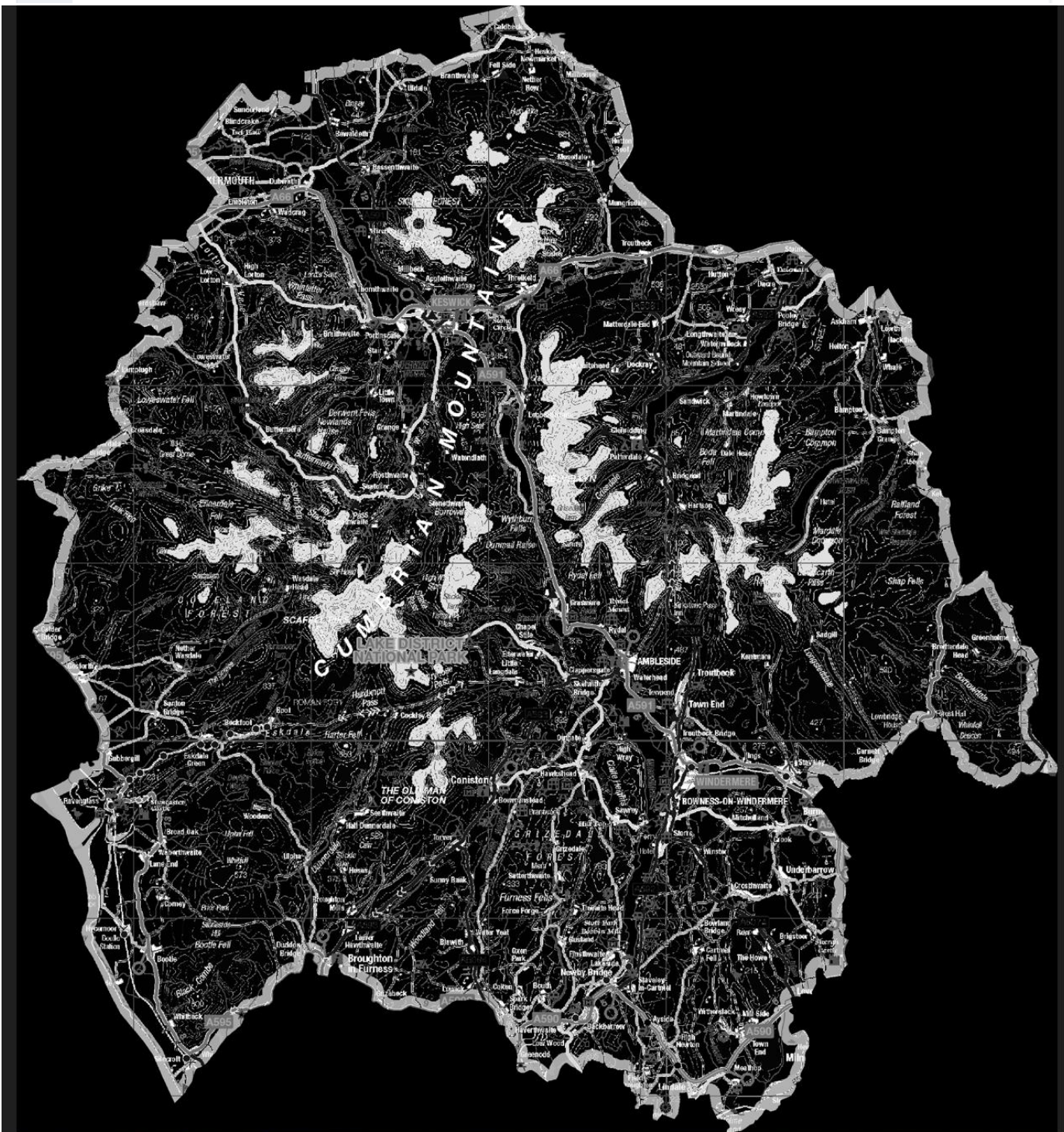
Query Query History

✓ 30

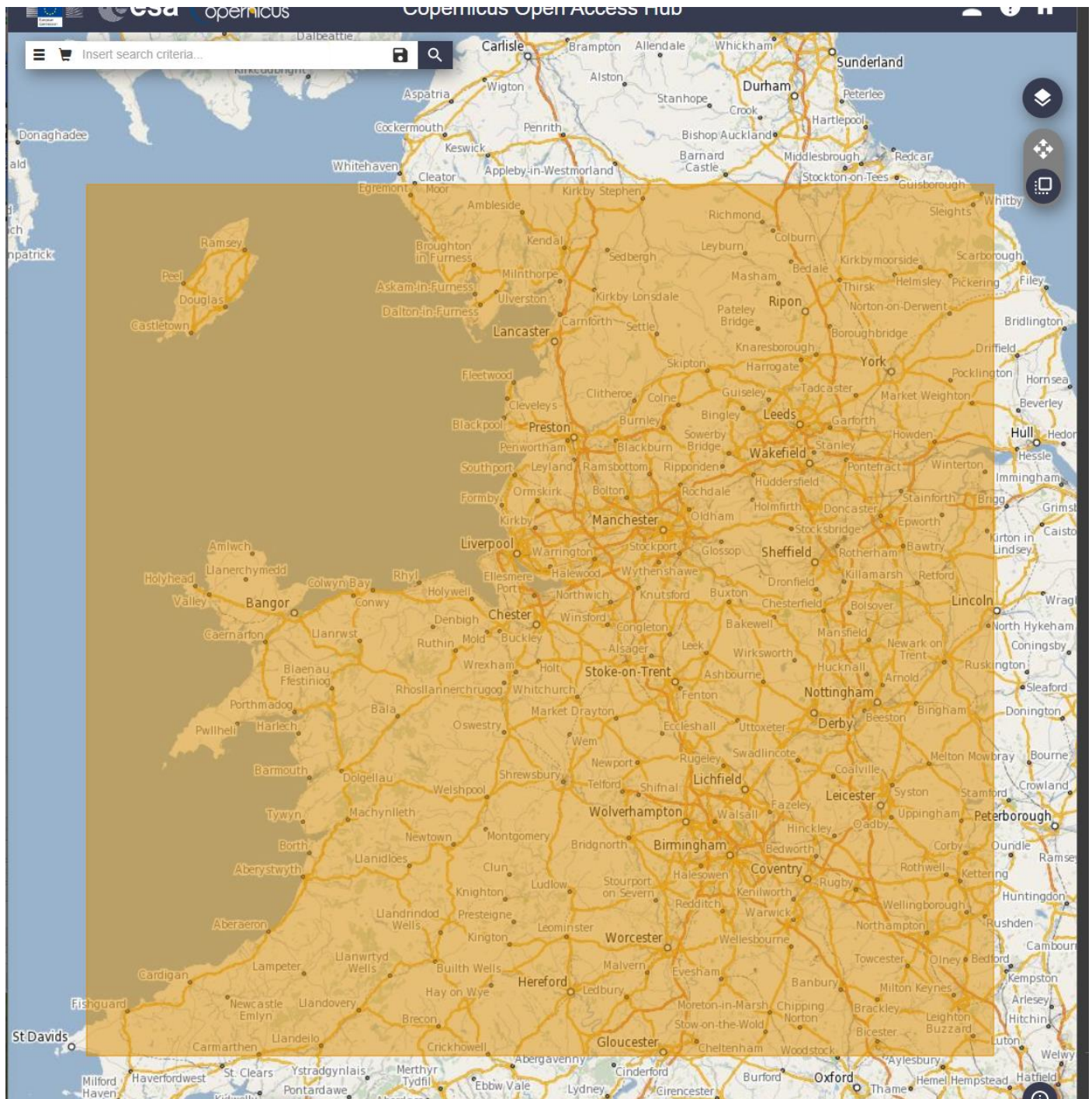
```

1 CREATE TABLE tmp_out_clipped AS
2 SELECT lo_from_bytea(0, ST_AsGDALRaster(ST_Union(st_union), 'GTiff', ARRAY['COMPRESS=
3 ) AS loid
4 FROM uk_lake_district;
5
6 SELECT lo_export(loid, 'D:\Semestr_7\BazyDanychPrzestrzennych\lab7\uk_lake_district.t
7 FROM tmp_out_clipped;
8
9 SELECT lo_unlink(loid)
10 FROM tmp_out_clipped;
11

```



8.



9.

raster2pgsql.exe -s 4277 -N -32767 -t 100x100 -l -C -M -d D:/Semestr_7/BazyDanychPrzestrzennych/lab7/satelite.jp2
sentinel | psql -d lab7 -h localhost -U postgres -p 5432

10.

Query Query History

```

1  create or replace function ndvi(
2  value double precision [] [] [],
3  pos integer [][],
4  variadic userargs text []
5  )
6  returns double precision as
7
8
9  begin
10
11  return (value [2][1][1] - value [1][1][1])/(value [2][1][1]+value
12  [1][1][1]); |
13  end;
14
15  language 'plpgsql' immutable cost 1000;
16  create table ndvi as
17  with r as(
18  select * from sentinel
19  )
20  select
21  r.rid, st_mapalgebra(
22  r.rast, array [1, 4],
23  'NDVI(double precision[], integer[], text[])'::regprocedure,
24  '32BF'::text
25  ) as rast
26  from r;
27  select * from uk_ld_sentinel;
28  select updaterrastersrid('sentinel', 'rast', 4277);
29  select st_srid(rast) from sentinel;
30
31  create table uk_ld_sentinel as
32  select a.rid, st_clip(a.rast, b.geom, true) as rast
33  from ndvi as a, national_parks as b
34  where b.id=1 and st_intersects(b.geom, a.rast);
35

```

at