

# “A User’s Guide to Importing Raster Data into PostGres & Tutorials on Raster Operations”

---

#Contributors:

Jerónimo Rodríguez

Jillian Long

Kevin Frech

Leonardo Hughes

//

RBB;

## Introduction

---

Why Raster?

Spatial data types provide the information that a computer requires to reconstruct the spatial data in digital form. In the raster world, we have grid cells representing real world features.

Raster data is made up of pixels (also referred to as grid cells). They are usually regularly-spaced and square but they don’t have to be. Rasters often look pixelated because each pixel has its own value or class.

### Advantages:

- Nature of data storage makes data analysis relatively easy and quick
- Useful for a wide range of applications (base maps, surface maps, thematic maps, etc)
- Advanced statistical use
- Can uniformly store points, lines, polygons and surfaces
- Great for continually varying information

### Disadvantages:

- Cell size determines resolution, resolution can be bad, which = can’t analyze data
- Hard to represent linear features
- Most output maps from grid-cell systems do not conform to high-quality cartographic needs
- Understanding raster data requires a bit of training the eye. This can be intimidating.
- The environment can work against you (ex: clouds in an image). Removing these while maintaining data integrity can be very frustrating (J. Rodriguez, 2018).

The raster2pgsql is a raster loading package which loads GDAL supported raster formats into sql suitable for loading into a PostGIS raster table.

Since the raster2pgsql is compiled as part of PostGIS, most often the raster types supported by raster2pgsql will be the same as those compiled in the GDAL dependency library.

You can use the -G switch to get a list of raster types supported by raster2pgsql

## Create a Raster Folder!

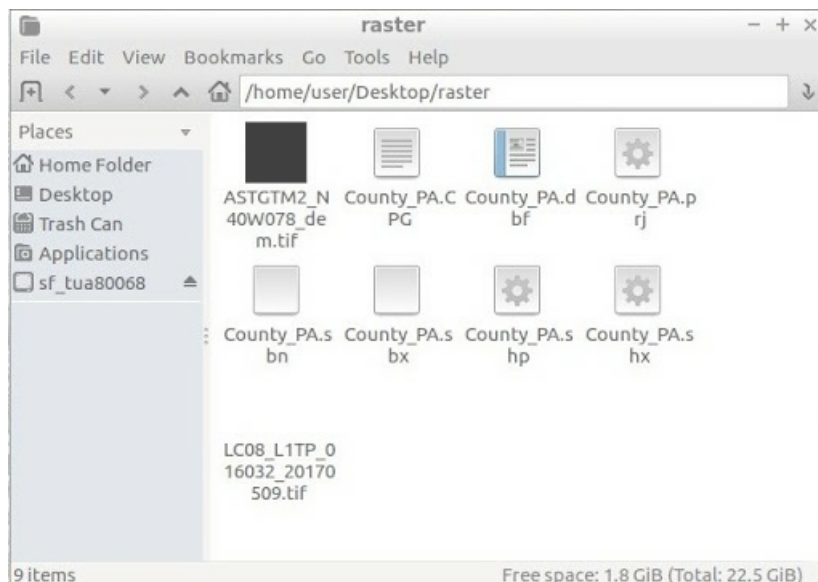
---

Download and unzip raster\_tutorial1.zip

Create folder /home/user/Desktop/raster

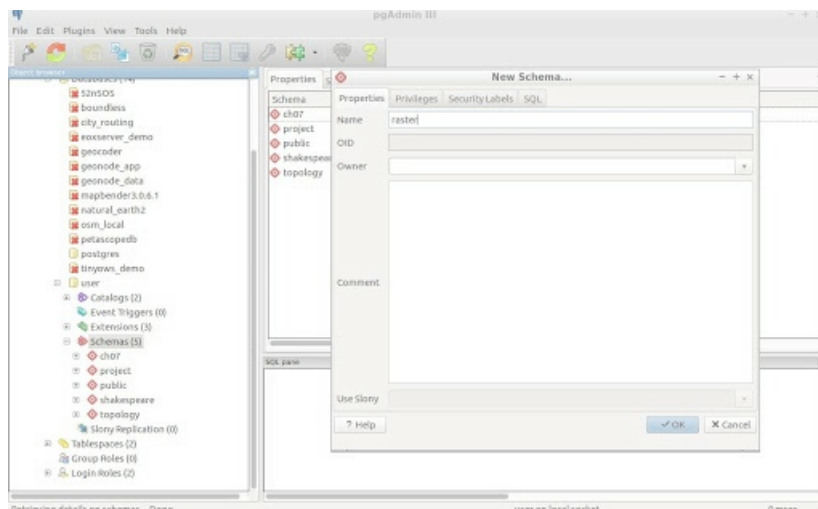
Place the following items into newly created folder:

- County\_PA
- ASTGTM2\_N40W078\_dem.tif
- LC08\_L1TP\_016032\_20170509.tif



## Create a Raster Schema!

---



# Linux

All aboard the Raster Train!

Initiate the Linux Command Line

```
LS
```

Before data import, it is good practice to inspect your data in order to be sure that the layers to be added into your database will in uniform Coordinate Reference Systems as well as similar extents. The `gdalinfo` tool can be used for this purpose.

For this tutorial, the RasterBadBoyz have pre-processed the two TIF files, see the appendix for further reading on data processing.

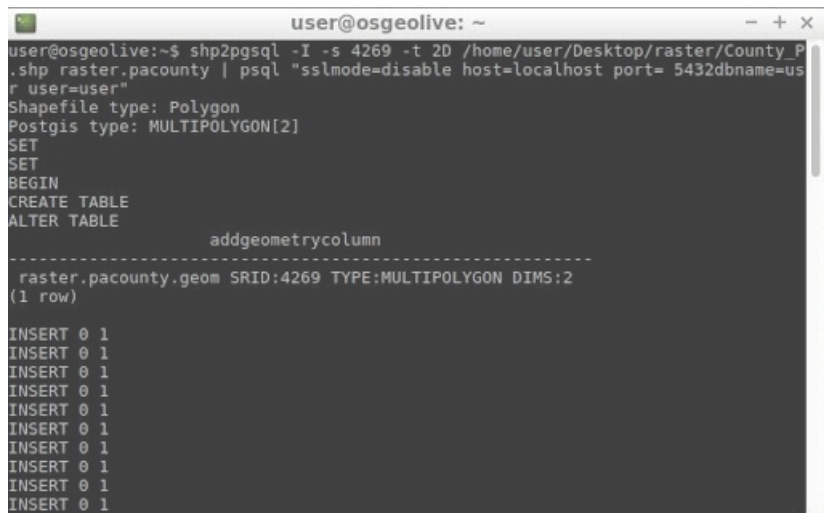
Set your working directory to your raster folder using the Linux command line!

```
cd /home/user/Desktop/raster
```

A terminal window titled 'user@osgeolive: ~/Desktop/raster'. The prompt is 'user@osgeolive:~\$'. The command 'cd /home/user/Desktop/raster' has been entered and executed. The prompt is now 'user@osgeolive:~/Desktop/raster\$'.

## vector pit stop

```
shp2pgsql -I -s 4269 -t 2D /home/user/Desktop/raster/County_PA.shp raster.pacounty | psql
```

A terminal window titled 'user@osgeolive: ~'. The prompt is 'user@osgeolive:~\$'. The command 'shp2pgsql -I -s 4269 -t 2D /home/user/Desktop/raster/County\_PA.shp raster.pacounty | psql "sslmode=disable host=localhost port= 5432 dbname=user user=user"' has been entered and executed. The output shows the shapefile type as Polygon, Postgis type as MULTIPOLYGON[2], and a series of SQL commands including SET, BEGIN, CREATE TABLE, ALTER TABLE, and multiple INSERT statements.

Cool, now lets implement `gdalinfo` to see the metadata.

```
gdalinfo ASTGMT2_N40W078_dem.tif
```

```
user@osgeolive: ~/Desktop/raster
user@osgeolive:~$ cd /home/user/Desktop/raster
user@osgeolive:~/Desktop/raster$ gdalinfo ASTGTM2_N40W078_dem.tif
Driver: GTiff/GeoTIFF
Files: ASTGTM2_N40W078_dem.tif
Size is 3601, 3601
Coordinate System is:
GEOGCS["WGS 84",
    DATUM["WGS 1984",
        SPHEROID["WGS 84",6378137,298.257223563,
            AUTHORITY["EPSG","7030"]],
        AUTHORITY["EPSG","6326"]],
        PRIMEM["Greenwich",0],
        UNIT["degree",0.0174532925199433],
        AUTHORITY["EPSG","4326"]]
Origin = (-78.0001388888888884,41.000138888888891)
Pixel Size = (0.000277777777778,-0.000277777777778)
Metadata:
  AREA_OR_POINT=Area
  TIFFTAG_DATETIME=2011:03:16 00:46:41
  TIFFTAG_DOCUMENTNAME=created at
  TIFFTAG_IMAGEDESCRIPTION=SILC TIFF
  TIFFTAG_RESOLUTIONUNIT=2 (pixels/inch)
  TIFFTAG_SOFTWARE=IDL 7.1.1, ITT Visual Information Solutions
  TIFFTAG_XRESOLUTION=100
```

gdalinfo LC08\_L1TP\_016032\_20170509.tif

```
user@osgeolive:~/Desktop/rasters$ gdalinfo LC08_L1TP_016032_20170509.tif
Driver: GTiff/GeoTIFF
Files: LC08_L1TP_016032_20170509.tif
       LC08_L1TP_016032_20170509.tif.aux.xml
Size is 3601, 3601
Coordinate System is:
GEOGCS["WGS 84",
    DATUM["WGS 1984",
        SPHEROID["WGS 84",6378137,298.257223563,
            AUTHORITY["EPSG","7030"]],
        AUTHORITY["EPSG","6326"]],
        PRIMEM["Greenwich",0],
        UNIT["degree",0.0174532925199433],
        AUTHORITY["EPSG","4326"]]
Origin = (-78.000138889999999,41.000138889999999)
Pixel Size = (0.000277777777778,-0.000277777777778)
Metadata:
  AREA_OR_POINT=Area
Image Structure Metadata:
  COMPRESSION=LZW
  INTERLEAVE=PIXEL
Corner Coordinates:
Upper Left  (-78.0001389, 41.0001389) ( 78d 0' 0.50"W, 41d 0' 0.50"N)
Lower Left  (-78.0001389, 39.9998611) ( 78d 0' 0.50"W, 39d59'59.50"N)
Upper Right (-76.9998611, 41.0001389) ( 76d59'59.50"W, 41d 0' 0.50"N)
Lower Right (-76.9998611, 39.9998611) ( 76d59'59.50"W, 39d59'59.50"N)
Center      (-77.5000000, 40.5000000) ( 77d30' 0.00"W, 40d30' 0.00"N)
Band 1 Block=3601x1 Type=Float32, ColorInterp=Gray
  Min=8762.822 Max=52445.584
  Minimum=8762.822, Maximum=52445.584, Mean=nan, StdDev=nan
  NoData Value=-3.39999999999999996e+38
  Metadata:
    STATISTICS_MAXIMUM=52445.50390625
    STATISTICS_MEAN=nan
    STATISTICS_MINIMUM=8762.822265625
    STATISTICS_STDDEV=nan
Band 2 Block=3601x1 Type=Float32, ColorInterp=Undefined
  Min=7730.429 Max=57955.121
  Minimum=7730.429, Maximum=57955.121, Mean=nan, StdDev=nan
  NoData Value=-3.39999999999999996e+38
  Metadata:
    STATISTICS_MAXIMUM=57955.12109375
    STATISTICS_MEAN=nan
    STATISTICS_MINIMUM=7730.4291992188
    STATISTICS_STDDEV=nan
```

Dummy Code **DO NOT TYPE**

```
raster2pgsql -s 4326 filepath.tif -t tiledimension -F schema.table| psql -d database
```

**DO TYPE**

Loading in the Digital Elevation Model (DEM) for study area using the raster2pgsl in the Linux command line!

```
raster2pgsql -s 4326 /home/user/Desktop/raster/ASTGTM2_N40W078_dem.tif -t 256x256 -F rast
```



```
user@osgeolive: ~/Desktop/raster
user@osgeolive:~/Desktop/raster$ raster2pgsql -s 4326 /home/user/Desktop/raster/ASTGTM2_N40W078_dem.tif -t 256x256 raster.demrast256 | psql -d user
Processing 1/1: /home/user/Desktop/raster/ASTGTM2_N40W078_dem.tif
BEGIN
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
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
```

Loading in the Landsat8 Multi-Band Raster for study area using the raster2pgsql in the Linux command line!

```
raster2pgsql -s 4326 /home/user/Desktop/raster/LC08_L1TP_016032_20170509.tif -t 256x256 r
```

```
user@osgeolive: ~/Desktop/raster
user@osgeolive:~/Desktop/raster$ raster2pgsql -s 4326 /home/user/Desktop/raster/LC08_L1TP_016032_20170509.tif -t 256x256 raster.landsat | psql -d user
Processing 1/1: /home/user/Desktop/raster/LC08_L1TP_016032_20170509.tif
BEGIN
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
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
```

Notes on Linux Ports

```
-d
```

Drops an existing table with the same name if there is one

```
-I
```

Adds a spatial index

```
-C
```

Enforces constraints - checks that data is registered properly

```
-M
```

Vacuum analyzes a table

```
-R
```

\*If included, then the actual values of the raster are not placed in database but rather 'registered' and then read from the original file on disk when required

```
-s 4326
```

The SRID (Coordinate Reference System), usually an EPSG code

```
-t 256X256
```

The number of pixels per tile

```
-F
```

Inserts filename

## Performing Spatial Queries

---

```
--- set searchpath
set search_path to raster, public;
show search_path;
```

Digital Elevation Model & Vector Data

```
--- return the metadata
SELECT rid, (foo.md).*
FROM (SELECT rid, ST_metadata(rast) AS md
FROM demrast256) AS foo;
```

Now onto some more advanced raster manipulations...

```
--- return stats of individual 256 resolution tiles

SELECT rid, (stats).*
FROM (SELECT rid, st_summarystats(rast) AS stats
FROM demrast256
) AS stats
;
```

What's a tile? 3600 by 3600? What did we just do?

```
--- return stats of individual 256 res tiles intersecting a county

SELECT rid, (stats).*
FROM (SELECT rid, ST_SUMMARYSTATS(rast) AS stats
FROM demrast256
JOIN pacounty ON
ST_INTERSECTS(demrast256.rast, ST_TRANSFORM(pacounty.geom, ST_SRID(demrast256.rast)))
WHERE pacounty.county_nam = 'MIFFLIN') AS stats
;
```

Query - user on local socket \*

File Edit Query Favourites Macros View Help

SQL Editor Graphical Query Builder

Previous queries

```

set search_path to public, raster;
--- return Summary Statistics for all tiles intersecting Mifflin County
SELECT (stats).*
FROM (SELECT ST_SUMMARYSTATS(ST_UNION(rast)) AS stats
FROM demrast256
JOIN pacounty ON
ST_INTERSECTS(demrast256.rast, ST_TRANSFORM(pacounty.geom, ST_SRID(demrast256.rast)))
WHERE pacounty.county_name = 'MIFFLIN') AS stats

```

Output pane

Data Output Explain Messages History

	rid	count	sum	mean	stdev	min	max
	integer	bigint	double precision	double precision	double precision	double precision	double precision
1	37	65536	36948935	472.243276874023	101.993127369375	307	677
2	38	65536	32395518	494.316375732422	78.7269718726919	288	664
3	39	65536	31237371	476.644454950855	96.3919226246043	261	683
4	40	65536	30461035	464.79858769043	117.345820287348	211	663
5	50	65536	35203014	537.167572021684	90.5900638668968	355	743
6	51	65536	38510091	485.54704204060	115.240240612110	221	709
7	52	65536	24809387	366.35417175293	142.809920343713	199	679
8	53	65536	28111040	432.00430453125	111.731229758399	224	699
9	54	65536	27440095	418.782623459961	131.732312256353	191	690
10	55	65536	26508057	314.83895508477	116.195459080825	186	634
11	64	65536	38231168	461.291815025	112.878333087675	262	736

OK. Unix Ln 11, Col 2, Ch 350 39 rows. 698 ms...

```

--- return UNIONED Summary Statistics for all tiles intersecting Mifflin County

SELECT (stats).*
FROM (SELECT ST_SUMMARYSTATS(ST_UNION(rast)) AS stats
FROM demrast256
JOIN pacounty ON
ST_INTERSECTS(demrast256.rast, ST_TRANSFORM(pacounty.geom, ST_SRID(demrast256.rast)))
WHERE pacounty.county_name = 'MIFFLIN'
) AS stats
;

```

Query - user on local socket \*

File Edit Query Favourites Macros View Help

SQL Editor Graphical Query Builder

Previous queries

```

set search_path to public, raster;
--- return UNIONED Summary Statistics for all tiles intersecting Mifflin County
SELECT (stats).*
FROM (SELECT ST_SUMMARYSTATS(ST_UNION(rast)) AS stats
FROM demrast256
JOIN pacounty ON
ST_INTERSECTS(demrast256.rast, ST_TRANSFORM(pacounty.geom, ST_SRID(demrast256.rast)))
WHERE pacounty.county_name = 'MIFFLIN'
) AS stats
;

```

Output pane

Data Output Explain Messages History

	count	sum	mean	stdev	min	max
	bigint	double precision	double precision	double precision	double precision	double precision
1	2555904	905516575	354.675517938076	138.212400687622	85	743

OK. Unix Ln 9, Col 28, Ch 249 1 row. 3.0 secs

```

--- clipped and unioned

SELECT (stats).*
FROM (SELECT ST_SUMMARYSTATS(st_union(st_clip(demrast256.rast, pacounty.geom))) AS stats
FROM demrast256
JOIN pacounty ON
ST_INTERSECTS(demrast256.rast, ST_TRANSFORM(pacounty.geom, ST_SRID(demrast256.rast)))
WHERE pacounty.county_name = 'MIFFLIN'
) AS stats
;

```

# Multi-Band Raster: Normalized Vegetation Index

---

```
set search_path to raster, public;
show search_path;

select *
from landsat
limit 10

SET postgis.gdal_enabled_drivers = 'ENABLE_ALL';
--- SET postgis.gdal_enabled_drivers = 'DISABLE_ALL';

SELECT short_name, long_name
FROM ST_GdalDrivers()
Order by short_name

SELECT rid, ST_NumBands(rast) AS numbands, filename AS filename
FROM landsat
LIMIT 10
;

--- select band metadata
SELECT rid , (foo.md).*
FROM (SELECT rid, ST_bandmetadata(rast) AS md
FROM demrast256 WHERE rid =2) AS foo
;

--- check the geometry type
SELECT ST_Geometrytype(geom) AS result
FROM pacounty
;

--- convert the SRID of county
ALTER TABLE pacounty
ALTER COLUMN geom TYPE
geometry(MultiPolygon,4326)
USING ST_Transform(geom,4326)
;

--- create a new table. necessary when creating to keep the primary key
CREATE TABLE raster.intersects AS
SELECT a.rast, b.COUNTY_NAM
FROM demrast256 AS a, pacounty AS b
WHERE ST_Intersects(a.rast, b.geom) AND b.COUNTY_NAM ILIKE 'mifflin'
;

--- Add serial primary key
ALTER TABLE raster.intersects
```



```

ADD COLUMN rid SERIAL PRIMARY KEY
;

--- add spatial index
CREATE INDEX idx_intersects_rast_gist ON raster.intersects
USING gist (ST_ConvexHull(rast));

--- add constraints
SELECT AddRasterConstraints('raster'::name, 'intersects'::name, 'rast'::name)
;

--- reconstituting the original files from the tiles
Create TABLE raster.dem_reconstituted AS
Select filename,
COUNT(rast) As num_tiles,
ST_union(rast) as rast
FROM demrast256
Group by filename;
alter table raster.dem_reconstituted
add column rid SERIAL PRIMARY KEY
;

CREATE INDEX idx_dem_reconstituted_rast_gist ON raster.dem_reconstituted
USING gist (ST_ConvexHull(rast));

SELECT AddRasterConstraints('raster'::name, 'dem_reconstituted'::name, 'rast'::name);

SELECT *
FROM dem_reconstituted

Create TABLE raster.intersects_rec AS
SELECT a.rast, b.COUNTY_NAM
FROM raster.dem_reconstituted AS a, raster.pacounty AS b
WHERE ST_Intersects(a.rast, b.geom) AND b.COUNTY_NAM ilike 'mifflin';
ALTER TABLE raster.intersects_rec
ADD COLUMN rid SERIAL PRIMARY KEY;

--- clip

CREATE TABLE raster.clip AS
SELECT ST_Clip(a.rast, b.geom, true), b.COUNTY_NAM
FROM raster.demrast256 AS a, raster.pacounty AS b
WHERE ST_Intersects(a.rast, b.geom) AND b.COUNTY_NAM LIKE 'MIFFLIN';

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

--- clip landsat

CREATE TABLE raster.landsat_mifflin AS
SELECT ST_Clip(a.rast, b.geom, true), b.COUNTY_NAM
FROM raster.landsat AS a, raster.pacounty AS b
WHERE ST_Intersects(a.rast, b.geom) AND b.COUNTY_NAM like 'MIFFLIN';

--- create primary key

```

```

ALTER TABLE raster.landsat_mifflin
ADD COLUMNrid SERIAL PRIMARY KEY;

--- add constraints
SELECT AddRasterConstraints('raster'::name, 'landsat_mifflin'::name, 'st_clip'::name);

SELECT *
FROM landsat_mifflin
LIMIT 10

--- map algebra. Calculate NDVI
CREATE TABLE raster.mifflin_ndvi AS
WITH r AS (
SELECT a.rid,st_clip(a.rast, b.geom,true) AS rast
FROM raster.landsat AS a, raster.pacounty AS b
WHERE b.county_nam ILIKE 'mifflin' AND ST_Intersects(b.geom,a.rast)
)
SELECT
r.rid,ST_MapAlgebra(
r.rast, 2,
r.rast, 5,
'([rast2.val] - [rast1.val]) / ([rast2.val] + [rast1.val])::float','32BF'
) AS st_clip
FROM r;

--- create primary key
ALTER TABLE raster.mifflin_ndvi
ADD COLUMN rid2 SERIAL PRIMARY KEY;

--- create index
CREATE INDEX idx_mifflin_ndvi_rast_gist ON raster.mifflin_ndvi
USING gist (ST_ConvexHull(st_clip));

SELECT *
FROM mifflin_ndvi

--- add constraints
SELECT AddRasterConstraints('raster'::name, 'mifflin_ndvi'::name, 'st_clip'::name);

```

fin.

## Additional Readings

---

PostGIS

[http://www.cef-cfr.ca/uploads/Membres/PierreRacine\\_FOSS4G\\_2011.pdf](http://www.cef-cfr.ca/uploads/Membres/PierreRacine_FOSS4G_2011.pdf)

<https://duncanjg.wordpress.com/2012/11/20/the-basics-of-postgis-raster/>\*\*

<https://www.systutorials.com/docs/linux/man/1-tiffcp/>

[http://blog.light42.com/wordpress/wp-content/uploads/2012/11/postgis\\_raster.pdf](http://blog.light42.com/wordpress/wp-content/uploads/2012/11/postgis_raster.pdf)

DEM Sources

<https://gisgeography.com/free-global-dem-data-sources/>

Landsat

<https://earthexplorer.usgs.gov/>