

OBJETIVOS

Información sobre cómo Giswater maneja los rásters, y como se pueden cargar tanto en un servidor LINUX como en un servidor Windows.

DESCRIPCIÓN

1 - TABLAS, VARIABLES Y FUNCIONES DEL SISTEMA

Las dos tablas que actúan sobre el sistema son:

ext_raster_dem

ext_cat_raster

Estas ya están creadas, pero el *ext_raster_dem* viene sin *restricciones*. En el caso de un esquema de **utils** existente, se almacenan allí, pero tienen nombres diferentes:

raster_dem

cat_raster

La tabla *raster_dem* se llena desde el exterior de acuerdo con el proceso explicado en el paso 2.

En caso de estar en un esquema corporativo de *utils*, la tabla de catálogo se llena automáticamente con un disparador *AFTER INSERT* en la toma de ráster:

file name (*rastercat_id*)

cur_user

tstamp

Por otro lado, para utilizar esta funcionalidad, existen dos variables:

SISTEMA: *admin_raster_dem* (debe ser *TRUE*)

USUARIO: *edit_insert_elevation_from_dem* (debe ser *TRUE*)

edit_update_elevation_from_dem (debe ser *TRUE*)

Los triggers de *node/conne* tanto en *Insert* como en *Update* en la captura automáticamente el valor de la cota. Por otro lado, la función **gw_fct_update_elevation_from_dem** de la caja de herramientas activa automáticamente la captura de todas las dimensiones para la capa seleccionada.

2 – CARGAR RÁSTER EN BD

Los conceptos claros a tener en cuenta son:

Nombre del fichero:

Se recomienda que el nombre incluya la mayor cantidad de información posible sobre el ráster ya que dará información en la metatabla de *ext_cat_raster* sobre el tipo que es:

dg_dem_2019_u48 (*proveedor, tipo de ráster, año de datos, hoja de mapa*)

De esta forma, cuando se inserta el ráster, también se llena el catálogo de ráster y lleva información detallada sobre el mismo.

Tipología de archivo:

Si todos los rásteres DEM se insertan en la misma tabla, todos deben ser iguales en términos de formato para que las restricciones de la columna de la tabla no se rompan.

En este sentido, al cargar el primer ráster, las restricciones deben crearse como se define en el punto dos de este documento.

Ráster almacenado dentro o fuera de bbdd

Dado que hay dos entornos O/S para la máquina donde se aloja PostgreSQL, este proceso se detalla para cada uno de los dos entornos.

Opción muy interesante para no cargar la base de datos y recargar archivos automáticamente (solo debes cambiar el archivo)

IMPORTANTE: SE PUEDE EJECUTAR EL PROCESO DESDE EL POSTGRES LOCAL O DESDE EL POSTGRES DEL SERVIDOR DÓNDE QUEREMOS INSERTAR EL RÁSTER, SIEMPRE QUE TENGAMOS ACCESO A ÉL CON VPN O OTRO MÉTODO. ES MÁS FÁCIL EN LOCAL.

ENTORNO WINDOWS

1. Verifique que haya un ejecutable raster2pgsql en la carpeta *bin* de PostgreSQL.
2. Abra el símbolo del sistema (cmd), vaya a la carpeta *bin* de PostgreSQL (cd C:\Program Files\PostgreSQL\11\bin\).
3. Ejecute el proceso como en el ejemplo, poniendo el SRID, la ruta al archivo, el tamaño del mosaico, el nombre de la tabla a la que se importa el ráster y la conexión a la base de datos:

```
raster2pgsql.exe -R -s 25831 -C -x raster.txt -t 1500x1500 -a utils.raster_dem -F -n rastercat_id |  
psql -d giswater -U postgres -p 5432
```

ENTORNO LINUX

Dependerá de la distribución, pero como regla general debe ser tal que:

Dado que PostgreSQL normalmente se instala en la ruta, la línea de comando puede ser totalmente sencilla:

```
raster2pgsql -s 25831 -C -x raster.txt -t 1500x1500 -a utils.raster_dem -F -n rastercat_id | psql -d  
giswater -U postgres -p 5432
```

Si por alguna razón las variables de entorno están deshabilitadas, deben habilitarse:

OPCIÓN A: archivo de *entorno* (con un servicio de recarga *postgresql*)

```
POSTGIS_ENABLE_OUTDB_RASTERS=1  
POSTGIS_GDAL_ENABLED_DRIVERS=ENABLE_ALL
```

OPCIÓN B: a través de la consola (mucho más fácil)

```
SET postgis.enable_outdb_rasters TO True;  
SET postgis.enabled_drivers TO enable_all;
```

ADVERTENCIA PARA EL USUARIO:

Si se hace con un usuario de PostgreSQL, este debe tener permisos de lectura para el archivo.
Si se hace con otro usuario (tipo root) este debe estar registrado en *pg_hba.conf* y en *SGDB*.

ANOTACIONES DE SENTENCIAS:

```
[-R] -s 25831 -C -x raster.txt -t 1500x1500 -a utils.raster_dem -F -n rastercat_id | psql -d giswater -U  
postgres -p 5432
```

[-R] (OPCIONAL) el ráster se almacena fuera de la base de datos. De lo contrario, se almacena en el interior.

ADVERTENCIA: El problema es que no es fácil trabajar con él. Puede ser el usuario del sistema y el usuario de Postgres debe ser el mismo y con permisos para leer / escribir en archivos.

Opcional y imprescindible en función de la estrategia de almacenamiento elegida.

NOTA: PARA ESTA OPCIÓN DEBE TENER LAS VARIABLES:

POSTGIS_ENABLE_OUTDB_RASTERS
POSTGIS_GDAL_ENABLED_DRIVERS=ENABLE_ALL

[-s 25831] SRID, **obligatorio**

[-C] *agregar restricciones*

Requerido solo al cargar el primer ráster. Las restricciones son:

raster height (number of rows in table), enforce_height_rast
raster width (number of columns in the raster): enforce_width_rast
value for no data: enforce_nodata_values_rast
number of bands (for dem 1) enforce_num_bands_rast
pixel type (1bit, 2bit, 4bit....) enforce_pixel_types_rast
set the scale for x: enforce_scalex_rast
set the scale for y: enforce_scaley_rast
set SRID: enforce_srid_rast
out_db (info maintenance outside database)
extension (enforce_max_extent_rast)

[-x] excluye la restricción de la dimensión espacial. **Obligatorio de usar si el propósito es poner más de un ráster en la misma tabla (que será lo habitual)**
extension (enforce_max_extent_rast)

[raster.txt]

Nombre del fichero
Sin espaciado, pero con metadatos.

[-t 1500x1500]

Tamaño de celda Bbdd.

Tiene límites - 5000x5000 - *Error sin memoria. Falló en la solicitud de tamaño.* El tamaño recomendado no debe exceder los 2000x2000 por fila.

Se creará una nueva tabla en la base de datos (no se permiten *actualizaciones*) con la estructura definida. El proceso divide el ráster en partes (según el tamaño definido), cada *fila* de la tabla es parte del ráster.

El punto clave es que el tamaño de la entrada 1500x1500 es un divisor del tamaño del ráster. Divisor ideal 1 a 1 pero si el ráster supera los 2000x2000 se debe dividir (siempre con divisores...). Ejemplos:

para ráster de 1000x1000 → t 1001x1001 (estará en 1 fila)
para ráster de 1000x1000 → t 1000x1000 (estará en 4 filas)
para ráster de 2000x2000 → t 2001x2001 (estará en 1 filas)

DOCUMENTO DE PROTOCOLO

P-27

Administrar Ráster

para ráster de 2200x2200 → t 1100x1100 (estará en 4 filas)

para ráster de 5555x5555 → t 1111x1111 (estará en 16 filas)

`[-a utils.raster_dem]` agrega ráster a la tabla, **obligatorio** ya que de lo contrario crearía uno nuevo con el conflicto que esto significa.

`[-F]` agrega el nombre del archivo, **obligatorio** e importante para conocer el nombre del archivo.

`[-n rastercat_id]` para el nombre de la columna donde insertar el nombre del archivo. **Obligatorio**

`[-d giswater -U postgres -p 5432]`

Parámetros de conexión: si se hace con un usuario de PostgreSQL, es directo. Si se hace con otro usuario, pedirá la contraseña que obviamente también se puede hacer...

PARA PROCESOS DE CARGA MASIVA

NOTAS

- Cuidado con las restricciones

PISTA: CARGAR EL TEMA DE RESTRICCIÓN PRIMERO Y LUEGO LOS DEMÁS

S/O

WINDOWS

```
for /r %%i in (*) do "%PG_PATH%\raster2pgsql.exe" -R -s 25830 -x %%i -t 1500x1500 -a utils.raster_dem -F -n rastercat_id | "%PG_PATH%\psql" -h 192.168.99.124 -d gis -U bgeoadmin
```

(Adjuntar un archivo bat)

LINUX

Pista:

El usuario (\$userSystem) debe ser el mismo que el que está conectado a la máquina, de lo contrario la emisión de la contraseña es necesario y se vuelve complicado (método peer, modificar pg_hba...)

Cuidado con el camino (ficheros & raster2pgsql)

Sentencia

TESTE:

```
for f in *.txt; do echo "Test $f"; done
```

EJECUCIÓN:

```
for f in *.txt; do /raster2pgsql -R -s 25831 -C -x $f -t 1500x1500 -a utils.raster_dem -F -n rastercat_id | psql -d giswater -U $userSystem -p 5432
```

REFERENCIAS

Documentación: https://postgis.net/docs/using_raster_dataman.html

DOCUMENTO DE PROTOCOLO**P-27***Administrar Ráster***FAQS**

¿Algunas queries de ejemplo de cargas de ráster directamente desde la línea de comandos de nuestro localhost?

- Insertamos en host 000.000.00:5432 con usuario 'admin' e insertamos directamente en una tabla existente 'utils.raster_dem' (variable -a):

```
"C:\Program Files\PostgreSQL\9.6\bin\raster2pgsql.exe" "C:\Users\usuari\Desktop\raster.tif" -I -C -x -a -s 25831 -t 1500x1500 -F -n rastercat_id utils.raster_dem | "C:\Program Files\PostgreSQL\9.6\bin\psql.exe" -h 000.000.00 -p 5432 -d gis -U admin
```

- Insertamos en host 000.000.00:5432 con usuario 'admin' e insertamos en una tabla nueva que se llamará 'ws.raster' (variable -c):

```
"C:\Program Files\PostgreSQL\11\bin\raster2pgsql.exe" "C:\Users\usuari\Desktop\mde\mde.tif" -I -C -x -c -s 25831 -t 1500x1500 -F -n rastercat_id ws.raster | "C:\Program Files\PostgreSQL\11\bin\psql.exe" -h 000.000.00 -p 5432 -d gis -U admin
```

REVISIONES

Acción	Usuario	Fecha
Creado	Barbara Rzepka	21/11/2019
Modificado	Xavier Torret	16/04/2020
Modificado	Xavier Torret	29/07/2020
Modificado	Xavier Torret	12/03/2021
Modificado	Xavier Torret	28/06/2021
Modificado	Albert Bofill	31/01/2022