



Rasters

Licence GNU FDL - Version 1.0



Plan

- Présentation
- Chargement
- Fonctions
- Exemples/TP



Rasters: Présentation

- Un seul type: RASTER = matrice de valeurs en **2D**
- Une table RASTER = une couverture
- Une ligne RASTER = une tuile ou un raster
- Représente une image, des valeurs physiques (altitude par ex), des données scientifiques...
- Stockage interne (WKB) ou externe (image)
- Pas de pyramides (nouvelle table pour cela)
- Nouveau:
 - Type: RASTER (cf.GEOMETRY/GEOGRAPHY)
 - Fonctions
 - tables de catalogue
 - outils de chargement (basés sur GDAL)



Rasters: Présentation

- Conversion entre geometry et raster
 - `ST_DumpToPolygons(raster)`
 - \Rightarrow geometry set
- Fonctions gérant les deux types:
 - `ST_Intersection(raster|geometry, raster|geometry, "raster"|"geometry")`
 - \Rightarrow raster/geometry
- Opérateurs « intelligents »:
 - `raster|geometry && raster|geometry`
 - \Rightarrow boolean
 - `ST_Intersects(raster|geometry, raster|geometry)`
 - \Rightarrow boolean



Rasters: Chargement dans PostGIS

- raster2pgsql (\Leftrightarrow shp2pgsql)
- Aide: raster2pgsql -? ou raster2pgsql
- Formats supportés: raster2pgsql -G (ceux de GDAL)
- Paramètres importants:
 - c|a|d|p: create, append, drop, prépare (comme shp2pgsql)
 - C: création des contraintes
 - I: création index spatial
 - s: srid
 - b: band
 - t: taille des tuiles (tiles)
 - R: stockage externe: seuls les métadonnées et le chemin sont stockés
 - l: création des overviews
 - N: valeur nodata



Rasters: Chargement dans PostGIS

- Chargement d'une couverture raster du monde
- Données: `tp/raster/TrueMarble.8km.5400x2700.tif`
- Table `monde_raster`
- Commande:
 - `raster2pgsql -M -I -C -Y -s 4326 -t 100x100 -l 2,4,8 TrueMarble.8km.5400x2700.tif monde_raster | psql`
- Les tables suivantes sont créées:
 - `monde_raster`, `o_2_monde_raster`, `o_4_monde_raster`, `o_8_monde_raster`
- Enregistrement dans le catalogue:
 - `select * from raster_columns;`



Rasters: Contraintes

- \d monde_raster

Indexes:

```
"monde_raster_pkey" PRIMARY KEY, btree (rid)
"monde_raster_rast_gist" gist (st_convexhull(rast))
```

Check constraints:

```
"enforce_height_rast" CHECK (st_height(rast) = 100)
"enforce_max_extent_rast" CHECK (st_coveredby(st_convexhull(rast), '0...0'::geometry))
"enforce_nodata_values_rast" CHECK
(_raster_constraint_nodata_values(rast)::numeric(16,10)[] =
'{NULL,NULL,NULL}'::numeric(16,10)[])
"enforce_num_bands_rast" CHECK (st_numbands(rast) = 3)
"enforce_out_db_rast" CHECK (_raster_constraint_out_db(rast) = '{f,f,f}'::boolean[])
"enforce_pixel_types_rast" CHECK (_raster_constraint_pixel_types(rast) = '{8BUI,
8BUI,8BUI}'::text[])
"enforce_same_alignment_rast" CHECK (st_samealignment(rast, '01...00'::raster))
"enforce_scalex_rast" CHECK (st_scalex(rast)::numeric(16,10) =
0.06666666666666667::numeric(16,10))
"enforce_scaley_rast" CHECK (st_scaley(rast)::numeric(16,10) =
(-0.06666666666666667)::numeric(16,10))
"enforce_srid_rast" CHECK (st_srid(rast) = 4326)
"enforce_width_rast" CHECK (st_width(rast) = 100)
```



Rasters: Catalogue

- **r_table_catalog**: non usité: nom de la base par défaut
- **r_table_schema**: le schéma contenant la table raster
- **r_table_name**: le nom de la table raster
- **r_raster_column** : colonne contenant le raster (plusieurs colonnes de type RASTER possibles)
- **Srid**: le SRID
- **scale_x** : facteur X entre pixel et coordonnées terrain
- **scale_y**: facteur X entre pixel et coordonnées terrain
- **blocksize_x**: largeur d'un bloc en pixel
- **blocksize_y** : hauteur d'un bloc en pixel



Rasters: Catalogue

- **same_alignment**: vrai si toutes les tuiles ont le même alignement (TODO)
- **regular_blocking**: information indicative: pas de recouvrement, alignement identique, taille des pixel, srid.
- **num_bands**: nombre de bandes
- **AddRasterConstraints** : tableau des types de pixel pour chaque bande
- **AddRasterConstraints** : tableau des valeurs de NODATA pour chaque bande
- **extent**: l'extension spatiale de la table. (en cas d'ajout de données dans la table, utiliser **DropRasterConstraints** puis **AddRasterConstraints** !). Cela permet la visu des données



Rasters: overviews

- Version dégradée du raster pour affichage à des résolutions moins grandes
- Créé automatiquement lors du chargement (option `-l`)
- Enregistrés dans un catalogue: `raster_overviews`:
- `o_table_catalog` : catalogue de la table (= nom de la BD)
- `o_table_schema`: schéma de la table
- `o_table_name`: nom de la table overview
- `o_raster_column`: nom de la colonne
- `r_*`: les informations relatives à la table source
- `overview_factor`: le facteur de réduction, en puissance de 2



Rasters: Fonctions

- **Gestion** (`AddRasterConstraint`, `dropRasterConstraint...`)
- **Constructeurs** (`st_AddBand`, `st_asRaster`, `st_makeEmptyRaster...`)
- **Accesseurs** (`st_height`, `st_numBands`, `st_rotation`)
- **Accesseurs par bande** (`st_bandMetaData`, `st_bandPixelType`, ...)
- **Accesseurs par pixel** (`st_pixelAsPolygon`, `st_value`, ...)
- **Editeurs** (`st_setRotation`, `st_setSRID`, `st_resample`, `st_snapToGrid`, ...)
- **Editeurs de bandes** (`ST_SetBandNoDataValue`, `ST_SetBandIsNoData`)
- **Analyse et stats** (`st_count`, `st_histogram`, `st_quantile`, ...)
- **Sorties** (`st_asBinary`, `st_asGDALRaster`, `st_asTIFF`, ...)
- **Processing** (`st_clip`, `st_hillShade`, `st_slope`, `st_mapAlgebraExpr`, ...)



Rasters: Fonctions

geomval	ST_NumBands	ST_BandIsNoData
histogram	ST_PixelHeight	ST_BandPath
raster	ST_PixelWidth	ST_BandPixelType
reclassarg	ST_ScaleX	ST_HasNoBand
summarystats	ST_ScaleY	ST_PixelAsPolygon
AddRasterConstraints	ST_Raster2WorldCoordX	ST_PixelAsPolygons
DropRasterConstraints	ST_Raster2WorldCoordY	ST_Value
PostGIS_Raster_Lib_Build_Date	ST_Rotation	ST_SetValue
PostGIS_Raster_Lib_Version	ST_SkewX	ST_SetGeoReference
ST_GDALDrivers	ST_SkewY	ST_SetRotation
ST_AddBand	ST_SRID	ST_SetScale
ST_AsRaster	ST_UpperLeftX	ST_SetSkew
ST_Band	ST_UpperLeftY	ST_SetSRID
ST_MakeEmptyRaster	ST_Width	ST_SetUpperLeft
ST_GeoReference	ST_World2RasterCoordX	
ST_Height	ST_World2RasterCoordY	
ST_MetaData	ST_IsEmpty	
	ST_BandMetaData	
	ST_BandNoDataValue	



Rasters: Fonctions

ST_Resample	ST_AsTIFF	ST_Union
ST_Rescale	Box3D	ST_Min4ma
ST_Reskew	ST_Clip	ST_Max4ma
ST_SnapToGrid	ST_ConvexHull	ST_Sum4ma
ST_Transform	ST_DumpAsPolygons	ST_Mean4ma
ST_SetBandNoDataValue	ST_Envelope	ST_Range4ma
	ST_HillShade	ST_Distinct4ma
ST_SetBandIsNoData	ST_Aspect	ST_StdDev4ma
ST_Count	ST_Slope	&&
ST_Histogram	ST_Intersection	&<
ST_Quantile	ST_MapAlgebraExpr	&>
ST_SummaryStats	ST_MapAlgebraExpr	ST_Intersects
ST_ValueCount	ST_MapAlgebraFct	ST_SameAlignment
ST_AsBinary	ST_MapAlgebraFct	
ST_AsGDALRaster	ST_MapAlgebraFctNgb	
ST_AsJPEG	ST_Polygon	
ST_AsPNG	ST_Reclass	



Exemples

- TP R1: introduction aux rasters
 - chargement d'une image dans PostGIS
 - vérification des données (contraintes, catalogue)
 - Affichage dans Qgis
- TP R2: extraction et sortie locale
 - extraction spatiale d'une partie du raster
 - reconstruction d'un raster complet (3 bandes)
 - écriture d'une partie du raster en PNG sur le disque
- TP R3: croisements spatiaux: profil altimétrique à partir d'un MNT (également TP webmapping)