

12

Traiter les données raster

Les rasters peuvent être modifiés avec PostGIS, soit pour en mettre à jour les propriétés, soit pour en modifier les données. Dans ce chapitre, nous verrons quelles opérations peuvent être menées pour simplement reparamétrer un raster et ainsi pouvoir le visualiser correctement. Nous aborderons également des traitements plus complexes pour faire parler les données qu'ils contiennent à l'aide notamment de fonctionnalités de calcul avancées utilisant le langage PL/pgSQL.

12.1. Modifier les rasters

Les rasters peuvent être modifiés à l'aide d'une simple requête SQL d'UPDATE sur une colonne raster.

Les modifications peuvent aussi bien concerner les propriétés du raster que les données en elles-mêmes. La mise à jour d'une colonne en SQL se passe toujours de la même façon : en remplaçant les valeurs de la colonne soit par une valeur apportée dans la requête, soit par une valeur calculée à partir de l'ancienne.

Au niveau des propriétés (sans modification des données)

Plusieurs fonctions permettent de modifier, de redéfinir les propriétés d'un raster. Attention, les données ne sont pas modifiées, seules les propriétés, c'est-à-dire les valeurs qui permettent de lire correctement le raster le sont.

Toutes les propriétés peuvent être redéfinies : le système de coordonnées avec `ST_SetSRID`, les coordonnées du point d'origine du raster (supérieur gauche) avec `ST_SetUpperLeft`, l'échelle de chaque pixel avec `ST_SetScale`, la distorsion avec `ST_SetSkew` et enfin la rotation avec `ST_SetRotation`.

Il est même possible de modifier toutes ces propriétés d'un seul coup avec `ST_SetGeoReference`.

Exemple 12.1 : *Redéfinir l'échelle d'un raster*

Prenons un raster de 10 × 10 pixels avec une échelle de 1 dans le système de coordonnées WGS 84 : chaque pixel mesure 1 degré de côté (pixel carré).

L'exemple sera créé dans une table dédiée :

```
CREATE TABLE exemple_raster.table_traitement (
  id serial PRIMARY KEY,
  commentaire text,
  rast raster
)
;
```

Création d'un raster vide :

```
INSERT INTO exemple_raster.table_traitement(
  commentaire,
  rast
)
VALUES
(
  'Raster echelle 1_1',
  ST_MakeEmptyRaster(
    10, -- Largeur en pixels
    10, -- Hauteur en pixels
    0, -- Coordonnée X de l'origine
    0, -- Coordonnée Y de l'origine
    1, -- Echelle selon l'axe des X
    1, -- Echelle selon l'axe des Y
    0, -- Distorsion selon l'axe des X
    0, -- Distorsion selon l'axe des Y
    4326 -- Système de coordonnées
  )
)
;
```

Ajout d'une bande avec la valeur 150 pour chaque pixel :

```
UPDATE
  exemple_raster.table_traitement
SET
  rast = ST_AddBand(
    rast, -- Colonne raster sur laquelle ajouter la bande
    1, -- Index de la bande
    '8BUI'::text, -- Type des pixels (8 bits non signés : de 0 à 255)
    150, -- Valeur initiale
    NULL -- Valeur sans données
  )
WHERE
  commentaire = 'Raster echelle 1_1'
;
```

Création de deux nouveaux rasters avec des échelles différentes à partir du raster précédent :

```

INSERT INTO exemple_raster.table_traitement(
  commentaire,
  rast
)
SELECT
  'Raster echelle 10_10',
  ST_SetScale(rast, 10)  -- Echelle redéfinie à 10
FROM
  exemple_raster.table_traitement
WHERE
  commentaire = 'Raster echelle 1_1'
UNION
SELECT
  'Raster echelle -1_-1',
  ST_SetScale(rast, -1)  -- Echelle redéfinie à -1
FROM
  exemple_raster.table_traitement
WHERE
  commentaire = 'Raster echelle 1_1'
;

```

La requête suivante permet de vérifier les redéfinitions de l'échelle :

```

SELECT
  id,
  Box3D(rast) as rast,      -- Contour du raster
  ST_Width(rast) AS largeur, -- Largeur
  ST_Height(rast) AS hauteur -- Hauteur
FROM
  exemple_raster.table_traitement
ORDER BY
  id
;

```

Le changement d'échelle ne modifie pas la taille du raster, il conserve un nombre de 10 × 10 pixels. En revanche son emprise change, il devient plus ou moins grand (en unité "terrain") que le raster original.

id	rast	largeur	hauteur
1	BOX3D(0 0 0,10 10 0)	10	10
2	BOX3D(0 0 0,100 100 0)	10	10
3	BOX3D(-10 -10 0,0 0 0)	10	10