

---

# PostGIS – Fonctions spatiales de base

---

**Nicolas Ribot - Licence GNU FDL - Version 1.2**

---

# Fonctions de gestion

---

- `AddGeometryColumn`: Ajoute une colonne géométrique a une table et met a jour les métadonnées spatiales
- `DropGeometryColumn`: Supprime une colonne géométrique d'une table spatiale et met a jour les métadonnées spatiales
- `DropGeometryTable`: Supprime une table spatiale et met a jour les métadonnées spatiales
- `postgis_full_version`: Affiche la version courante de Postgis, GEOS, PROJ4 et PostgreSQL
- `UpdateGeometrySRID`: Met à jour le SRID de toutes les entités d'une colonne géographique, de la table `geometry_columns`, et de la contrainte sur le SRID de la table



---

# Constructeurs de géométries

---

- `ST_GeomFromText` : Construit une géométrie a partir de sa représentation WKT.
  - Exemple:
    - `select st_geomFromText('2.5 43', 4326)`
- `ST_GeometryFromText` : Alias pour `st_geomFromText`
- `ST_GeomFromWKB` : Construit une géométrie a partir de sa représentation WKB.
- `ST_Box2D`, `ST_Box3D` : Construit une bbox a partir de points.
- `ST_MakePoint`/`ST_Point` : Construit un point à partir de coordonnées
  - Exemple:
    - `SELECT ST_MakePoint(1,2);`



- ST\_MakeLine : Construit une ligne a partir de points
  - Exemple:
    - `SELECT ST_AsText(ST_MakeLine(ST_MakePoint(1,2), ST_MakePoint(3,4))) ;`
- ST\_MakePolygon : Construit un polygone a partir d'une ligne fermée
  - Exemple:
    - `SELECT ST_AsText(ST_MakeLine(ST_MakePoint(1,2), ST_MakePoint(3,4))) ;`

---

# Accesseurs de géométries

---

- GeometryType/ST\_GeometryType: Retourne le type de la géométrie ('POINT', 'POLYGON'), respectivement ('ST\_POINT', 'ST\_POLYGON')
- ST\_Boundary: Retourne l'ensemble formant la frontière d'un objet

- Exemple:

- `select st_boundary(st_point(0, 0));`
- `select st_boundary(st_geomFromText('LINESTRING(0 0, 1 1, 2 2)', -1));`
- `select astext (st_boundary (st_polygonFromText ('POLYGON((0 0, 6 0, 6 6, 0 6, 0 0), (1 1, 5 1, 5 5, 1 5, 1 1))')));`



- `st_CoordDim`: Retourne la dimension des coordonnées de la géométrie
- `ST_EndPoint/ST_StartPoint`: Retourne le dernier, respectivement le premier point d'une LINESTRING
- `ST_Envelope`: Retourne une géométrie représentant le rectangle englobant de la géométrie.
- `ST_ExteriorRing`: Retourne une LINESTRING représentant l'extérieur du POLYGONE
- `ST_GeometryN`: Retourne la Nieme géométrie (commençant à 1) d'une collection (GeomCol, MULTI\*)
- `ST_InteriorRingN`: Retourne la Nieme LINESTRING représentant l'intérieur d'un POLYGONE



- `ST_IsClosed`: Vrai si le premier et le dernier point d'un **LINESTRING** sont les mêmes
- `ST_IsEmpty`: Vrai si cette géométrie est vide (`GEOMETRYCOLLECTION(EMPTY)`)
- `ST_IsRing`: Retourne vrai si la géométrie est un ring (linestring fermée)
- `ST_IsSimple`: Retourne vrai si la géométrie est simple (polygone ne se croisant pas, par exemple)
- `ST_IsValid`: Vrai si cette géométrie est bien formée
- `ST_IsValidReason`: Retourne la raison de l'invalidité de la géométrie. (v 1.4+)



- `ST_NPoints`: Retourne le nombre total de sommets dans la géométrie.
- `ST_NRings`: Si la géométrie est un POLYGONE ou MULTIPOLYGON, retourne le nombre total de rings.
- `ST_NumGeometries`: Si la géométrie est une GEOMETRYCOLLECTION ou un MULTI\*, retourne le nombre total de géométries dans la collection.
- `ST_NumInteriorRings`: Si la géométrie est un POLYGONE ou un MULTIPOLYGON, retourne le nombre de rings intérieurs dans le **premier polygone** de la géométrie.



- `ST_PointN`: Retourne le Nieme point dans une `ST_LINESTRING` ou `ST_CIRCULARSTRING`.
- `ST_Summary`: Retourne un résumé de la géométrie
  - **Exemple:**
    - `select st_summary(the_geom) from departements;`
- `ST_X`, `ST_Y`, `ST_Z`: Retourne les coordonnées X, Y, Z, respectivement, du `POINT`



---

## Editeurs de géométries

---

- `ST_AddPoint/ST_RemovePoint`: Ajoute/supprime un point à une **LINESTRING** avant le/au point de position donnée (0-based index).
- `ST_Affine, ST_Rotate, ST_RotateX, ST_RotateY, ST_Scale, ST_Transcale`: Effectue des opérations math. de transformation affine, rotation mise à l'échelle, translation, etc.
- `ST_Force2D, ST_Force3D, ST_Force3DM`: Force la dimension de la géométrie
- `ST_ForceRHR`: Force l'orientation des sommets du **POLYGON** selon la règle de la main droite (sens horaire)



- ST\_LineMerge : Assemble des LINESTRING entre elles provenant d'une MULTILINESTRING
- ST\_Multi : Force la géométrie à être de la forme MULTI\*.
  - Exemple st\_multi(point) -> MULTIPOINT
- ST\_Force2D, ST\_Force3D, ST\_Force3DM : Force la dimension de la géométrie
- ST\_ForceRHR : Force l'orientation des sommets du POLYGONE selon la règle de la main droite (sens horaire)
- ST\_Reverse : Inverse l'ordre des sommets de la géométrie



# Editeurs de géométries (suite)



- ST\_Transform: Change le système de coordonnées de la géométrie passée en argument et retourne une nouvelle géométrie
  - Exemple:
    - `ST_Transform(geometry geom, integer srid)`
    - `SELECT ST_Transform(the_geom, 4326) FROM ma_table;`
- ST\_SetSRID: Met à jour le SRID de la géométrie
- ST\_Segmentize: Retourne la geometry modifiée de telle sorte que la longueur maximum d'un segment soit inférieur ou égale a la distance donnée.





---

## Sorties en différents formats

---

- `ST_Binary` : Retourne la représentation WKB de la géométrie
- `ST_AsGeoJson` : Retourne la représentation GeoJson de la géométrie
- `ST_AsGML` : Retourne la représentation GML 2 ou 3 de la géométrie
- `ST_AsKML` : Retourne la représentation KML de la géométrie, avec contrôle de la sortie (précision, version)
- `ST_AsSVG` : Retourne la représentation SVG (data path) de la géométrie
- `ST_AsGeoHash` : Retourne la représentation GeoHash de la géométrie



---

# Relations spatiales et mesures

---

- `ST_Azimuth`: Retourne l'angle en radian formé par le vecteur composé des deux POINTs
- `ST_LineCrossingDirection`: Etant données deux LINESTRING, retourne un nombre entre -3 et 3 donnant le type de croisement, 0 voulant dire pas de croisement:
  - 0: LINE NO CROSS
  - -1: LINE CROSS LEFT
  - 1: LINE CROSS RIGHT
  - -2: LINE MULTICROSS END LEFT
  - 2: LINE MULTICROSS END RIGHT
  - -3: LINE MULTICROSS END SAME FIRST LEFT
  - 3: LINE MULTICROSS END SAME FIRST RIGHT



- `ST_Distance`: Retourne la distance cartésienne minimale entre deux géométries.
- `ST_MaxDistance`: Retourne la distance cartésienne maximale entre deux géométries.
- `ST_DistanceSphere`: Retourne la distance en mètres de deux points situés sur la sphère terrestre (rayon 6370986 m)
- `ST_DistanceSpheroid`: Retourne la distance en mètres de deux points situés sur le sphéroïde donné
- `ST_DWithin`: Retourne vrai si deux géométries sont plus proche que la distance donnée.



- `ST_OrderingEquals` : Retourne vrai si les deux géométries sont égales et leurs points sont dans le meme ordre
- `ST_Relate` : Retourne vrai si les deux géométries ont une relation spatiale entre leurs intérieurs, extérieurs, frontières correspondant a la définition données dans la matrice d'intersection. Si aucune matrice d'intersection n'est passée, retourne la matrice d'intersection maximale qui relie les deux géométries



# Relations spatiales : st\_relate

- Matrice d'intersection (DE-9IM) Dimensionally-extended, 9 intersection matrix

	Interior	Boundary	Exterior
Interior	$\dim( I(a) \cap I(b) )$	$\dim( I(a) \cap B(b) )$	$\dim( I(a) \cap E(b) )$
Boundary	$\dim( B(a) \cap I(b) )$	$\dim( B(a) \cap B(b) )$	$\dim( B(a) \cap E(b) )$
Exterior	$\dim( E(a) \cap I(b) )$	$\dim( E(a) \cap B(b) )$	$\dim( E(a) \cap E(b) )$

Where:

T == {0,1,2}

F == empty set

\* == don't care

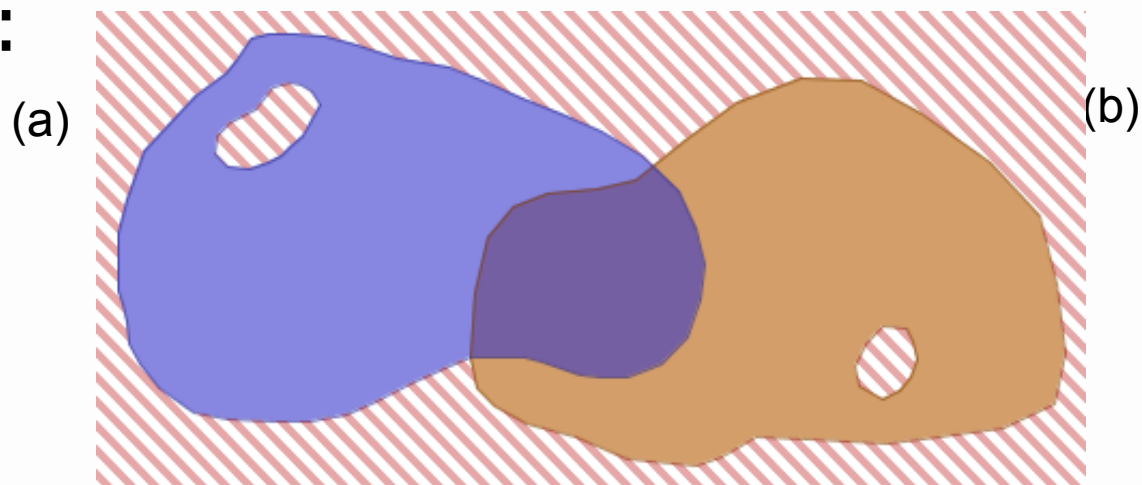
0 == dimensional 0 - point

1 == dimensional 1 - line

2 == dimensional 2 - area

# Relations spatiales : st\_relate

- Exemple:



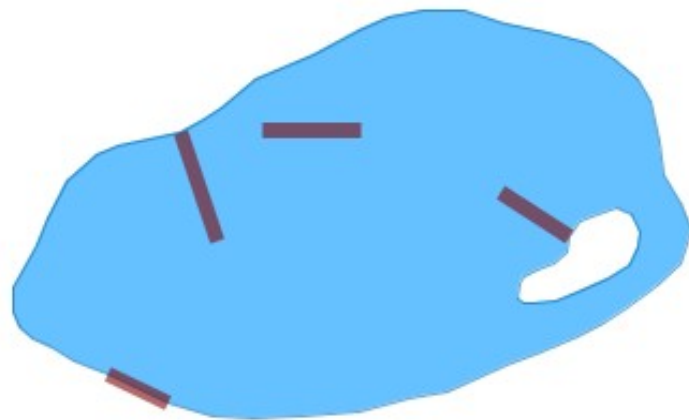
**Interior    Boundary    Exterior**

<b>Interior</b>	2	1	2
<b>Boundary</b>	1	0	1
<b>Exterior</b>	2	1	2

ST\_Relate(a, b) = '212101212'



- Exemple: Trouver tous les pontons inclus dans le lac, ne touchant pas les berges



Quels prédicats utiliser ? ST\_Within? ST\_Contains?  
ST\_Touches?

```
SELECT a.id
FROM docks a, lakes b
WHERE a.geom && b.geom
AND ST_Relate(a.geom, b.geom, 'TFFTFF212');
```

- Renvoie la surface en unités carrées (\*) d'une géométrie polygonale

- Prototype:

```
ST_Area(geometry geom)
```

- Exemple:

```
SELECT ST_Area(the_geom) FROM ma_table;
```

- Voir aussi:

- ST\_Perimeter

- Renvoie la géométrie du centroïde d'une géométrie polygonale
- Prototype:  
`ST_Centroid(geometry geom)`
- Exemple:  
`SELECT ST_Centroid(the_geom) FROM ma_table;`
- Attention, le centroïde n'est pas toujours situé sur la surface
- Voir aussi:
  - `ST_PointOnSurface`

- Renvoie la longueur(\*) d'une géométrie linéaire

- Prototype:

```
ST_Length(geometry geom)
```

- Exemple:

```
SELECT ST_Length(the_geom) FROM ma_table;
```

- Voir aussi:

- ST\_Length\_Sphere/spheroid (distance en mètres)
- ST\_Perimeter

(\*) Distance en unité des données, dépendant du système de projection.

- Renvoie la BBOX **agrégée** d'un ensemble de géométries

- Prototype:

```
ST_Extent(geometry_set geom)
```

- Exemple:

```
SELECT ST_Extent(the_geom) FROM ma_table;
```

- Voir aussi:

- ST\_Box2D et ST\_Box3D
- ST\_Expand
- ST\_Enveloppe
- ST\_Estimated\_Extent



# Fonction de traitement géométrique

- Renvoie une géométrie généralisée (algo: Douglas Peuker)

- Prototype:

```
ST_Simplify(geometry geom, integer tolerance)
```

- Exemple:

```
SELECT ST_Simplify(the_geom, 100) FROM ma_table;
```

- Voir aussi:

- ST\_SnapToGrid
- ST\_SimplifyPreserveTopology



- Renvoie la géométrie résultant de l'intersection entre deux géométries données

- Prototype:

```
ST_Intersection(geometry geom_A, geometry geom_B)
```

- Exemple:

```
SELECT ST_Intersection(
    (SELECT the_geom FROM ma_table WHERE gid='4'),
    (SELECT the_geom FROM ma_table WHERE gid='2')
);
```

- Voir aussi:

- ST\_Union
- ST\_Difference, ST\_sym\_difference



- Renvoie une géométrie représentant un buffer de X unités(\*) autour d'une géométrie donnée. (unités des données)

- Prototype:

```
ST_Buffer(geometry geom, double buffer_distance,  
          [integer number_of_segments])
```

- Exemple:

```
SELECT ST_Buffer(the_geom, 30) FROM ma_table;
```

- Nota:

- La valeur du buffer peut être négative (buffer inverse)

- `ST_Collect` : Retourne une géométrie de type collection (`GeometryCollection` ou `MULTI*`) à partir de la liste de géométries donnée (fonctionne avec une colonne, deux géométries ou un tableau de géométries).
- `ST_ConvexHull` : Retourne l'enveloppe convexe de la géométries donnée
- `ST_Dump` : Retourne un ensemble de géométries constituant la géométrie donnée (Collection)
- `ST_Polygonize` : Retourne une `GEOMETRYCOLLECTION` contenant des polygones possiblement formés à partir du réseau de lignes constitué par la géométrie donnée.

-

