Formation PostGIS

Vecteurs – analyse et structure / gestion des tables - Octobre 2020

Fabien Guerreiro

INSTITUT NATIONAL SUPÉRIEUR DES SCIENCES AGRONOMIQUES, DE L'ALIMENTATION ET DE L'ENVIRONNEMENT

MEMBRE DE agreenium l'institut agronomique, vétérinaire & forestier de France

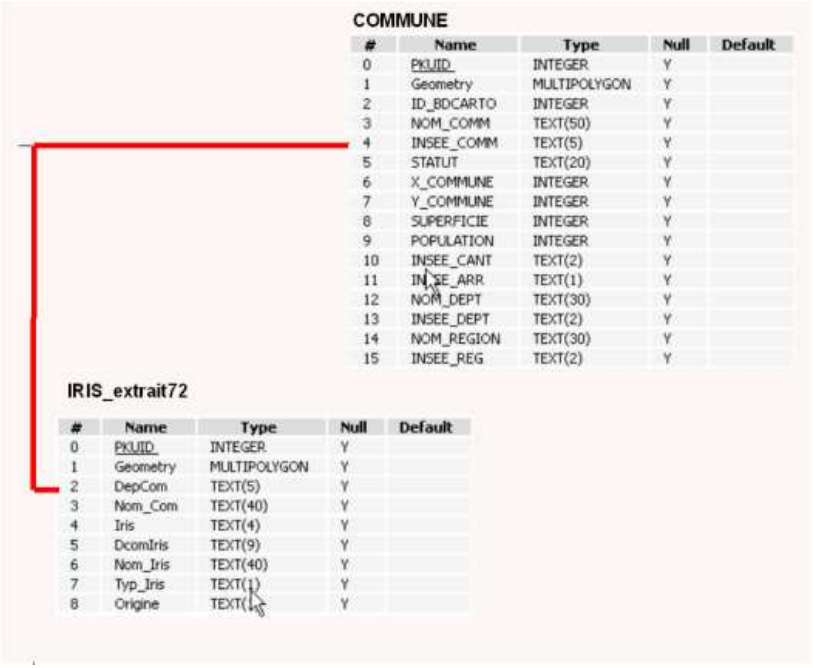
UBFC UNIVERSITÉ BOURGOGNE FRANCHE-COMTÉ

Jointures attributaires

Les requêtes

- … WHERE NomTable1.NomColonne1 = NomTable2.NomColonne2

- SELECT <colonnes> FROM <table1> JOIN <table2> ON <condition de jointure>



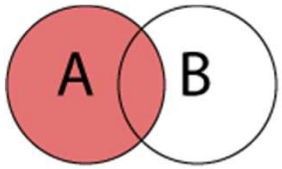
Exercice

A l'aide du constructeur graphique de requête, rechercher les IRIS des communes simples, et trier le résultat par nom d'IRIS.
Doivent apparaître dans le résultat, la géométrie des IRIS, leur nom, le nom de leur commune, et le nom de leur département.

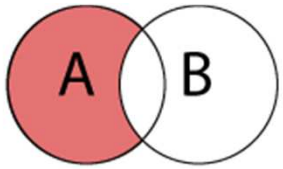
Jointures attributaires

Les requêtes

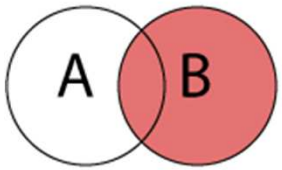
- SELECT <colonnes> FROM <table1> JOIN <table2> ON <condition de jointure>



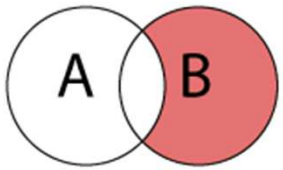
LEFT JOIN
SELECT *
FROM A
LEFT JOIN B ON A.key = B.key;



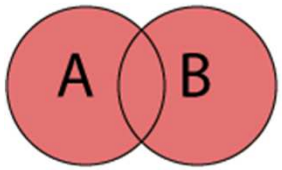
LEFT JOIN moins intersection B
SELECT *
FROM A
LEFT JOIN B ON A.key = B.key
WHERE B.key IS NULL;



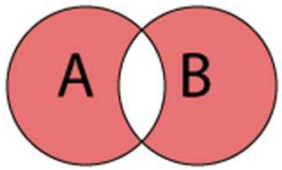
RIGHT JOIN
SELECT *
FROM A
RIGHT JOIN B ON A.key = B.key;



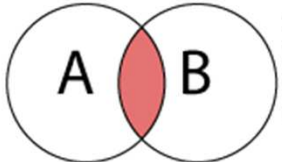
RIGHT JOIN moins intersection A
SELECT *
FROM A
RIGHT JOIN B ON A.key = B.key
WHERE A.key IS NULL;



FULL JOIN
SELECT *
FROM A
FULL OUTER JOIN B ON A.key = B.key;



FULL JOIN moins l'intersection
SELECT *
FROM A
FULL OUTER JOIN B ON A.key = B.key
WHERE A.key IS NULL
OR B.key IS NULL;



INNER JOIN
SELECT *
FROM A
INNER JOIN B ON A.key = B.key;

Jointures spatiales

Comparaison

ST_Equals(geometry A, geometry B)
ST_Intersects(geometry A, geometry B)
ST_Disjoint(geometry A, geometry B)
ST_Crosses(geometry A, geometry B)
ST_Overlaps(geometry A, geometry B)
ST_Touches(geometry A, geometry B)
ST_Within(geometry A, geometry B
ST_Contains(geometry A, geometry B)

Les fonctions suivantes sont également intéressantes :
ST_Dwithin(geometry A, geometry B, distance)
ST_Distance(geometry A, geometry B)

Exemple :
SELECT id_a, table_a.geom FROM table_a, table_b WHERE ST_Intersects(table_a.geom, table_b.geom);
Renvoit les objets de table_a qui intersectent ceux de table_b, sans les découper.

Géométrie dans les agrégations

GROUP BY

Toutes les colonnes en sortie (sauf celle du critère de regroupement) doivent faire l'objet d'une fonction d'agrégation, ce doit être également le cas pour la géométrie :

- **ST_UNION(geom)**
- **ST_Multi()**
- **ST_Linemerge()**

QGIS demande un identifiant unique, à recréer avec la requête :

- **row_number() over()**

ST_Intersection(geometry A, geometry B)

Opération sur les géométries

Exemple :

SELECT id_a, ST_Intersection(table_a.geom, table_b.geom) AS geom
FROM table_a, table_b;

Découpe les objets de table_a pour ne garder que ceux qui intersectent table_b.

Exercice pratique

1. Quels sont les ponctuels hydrographiques de la commune de La Flèche ?

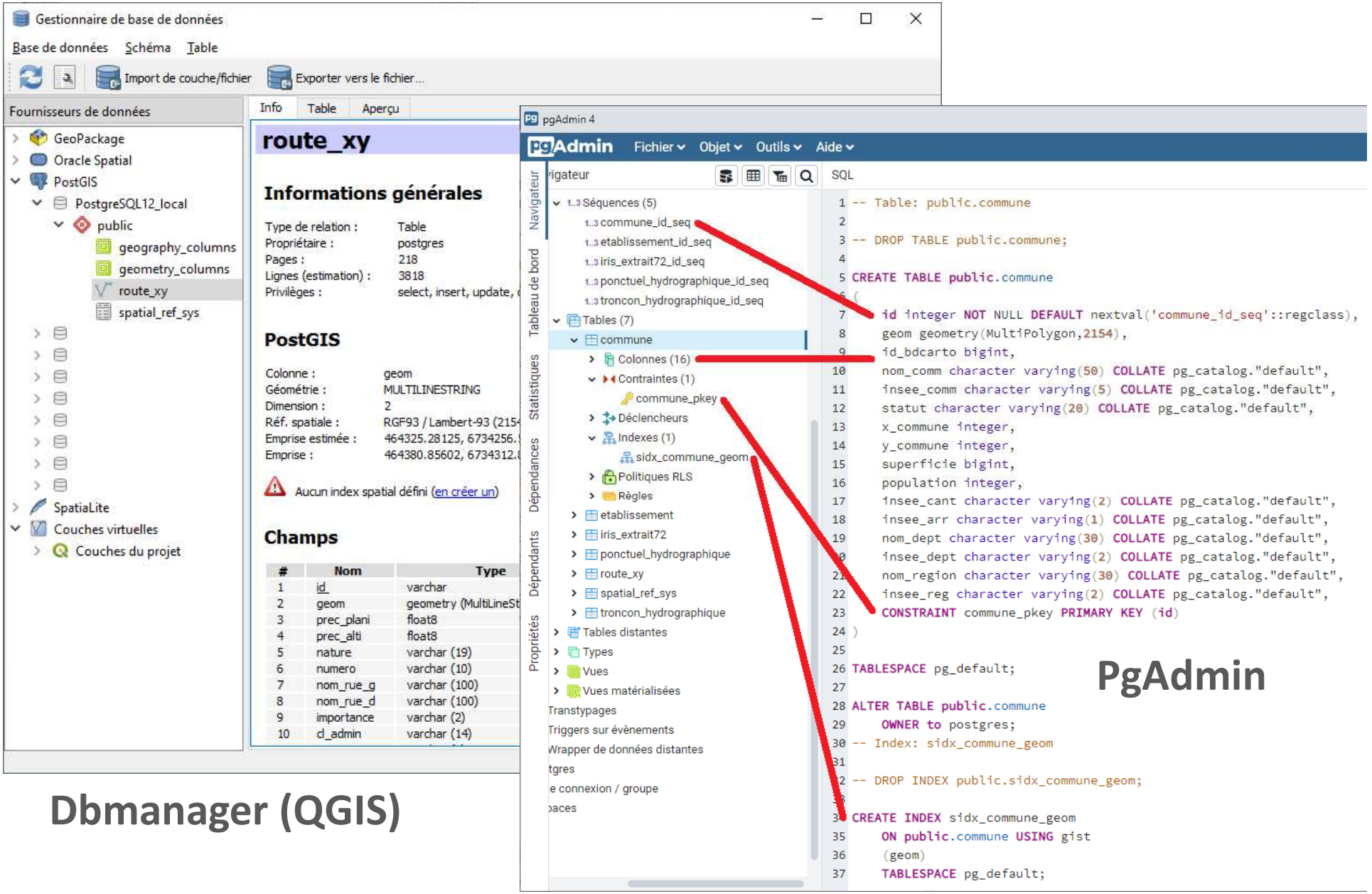
2. Quelle est la longueur de la 'rivière le loir' dans chacune des communes intersectées par le cours d'eau ?

3. Sélectionner les 'PONCTUELS HYDROGRAPHIQUES' qui sont à moins de 5 km d'un établissement d'enseignement (couche ETABLISSEMENT)

4. Quel est l'établissement le plus proche du centroide de la commune de la Flèche?

On utilisera les coordonnées X_COMMUNE et Y_COMMUNE et la fonction **ST_Makepoint()**. Le SRID (Identifiant du Système de Référence Spatial) est 2154, mais on pourra le cas échéant généraliser la requête à tout SRID en utilisant la fonction **ST_Srid()** qui récupère le srid d'une géométrie.

Définition d'une table



Dbmanager (QGIS)

PgAdmin

Définition SQL d'une table

Base

CREATE TABLE commune
(id integer NOT NULL, id_bdcarto integer, [...]);

Séquence

DEFAULT nextval('"COMMUNE_id_seq"'::regclass)

Clé primaire

ALTER TABLE commune
ADD CONSTRAINT "COMMUNE_pkey" PRIMARY KEY(id);

Droits

ALTER TABLE commune OWNER TO postgres;

Géométrie

ALTER TABLE commune ADD COLUMN geom geometry(MultiPolygon,2154);

La géométrie d'une table

La table spatial_ref_sys

Références EPSG

	srid [PK] integer	auth_name character varying (256)	auth_srid integer	srtext character varying (2048)	proj4text character varying (2048)
1	2000	EPSG	2000	PROJCS["Anguilla 1957 / Britis...	+proj=tmerc +lat_0=0 +lon_0=-6...
2	2001	EPSG	2001	PROJCS["Antigua 1943 / British...	+proj=tmerc +lat_0=0 +lon_0=-6...
3	2002	EPSG	2002	PROJCS["Dominica 1945 / Briti...	+proj=tmerc +lat_0=0 +lon_0=-6...
4	2003	EPSG	2003	PROJCS["Grenada 1953 / Britis...	+proj=tmerc +lat_0=0 +lon_0=-6...
5	2004	EPSG	2004	PROJCS["Montserrat 1958 / Bri...	+proj=tmerc +lat_0=0 +lon_0=-6...
6	2005	EPSG	2005	PROJCS["St. Kitts 1955 / British...	+proj=tmerc +lat_0=0 +lon_0=-6...
7	2006	EPSG	2006	PROJCS["St. Lucia 1955 / Britis...	+proj=tmerc +lat_0=0 +lon_0=-6...
8	2007	EPSG	2007	PROJCS["St. Vincent 45 / Britis...	+proj=tmerc +lat_0=0 +lon_0=-6...
9	2008	EPSG	2008	PROJCS["NAD27(CGQ77) / SCo...	+proj=tmerc +lat_0=0 +lon_0=-5...
10	2009	EPSG	2009	PROJCS["NAD27(CGQ77) / SCo...	+proj=tmerc +lat_0=0 +lon_0=-5...
11	2010	EPSG	2010	PROJCS["NAD27(CGQ77) / SCo...	+proj=tmerc +lat_0=0 +lon_0=-6...

La vue geometry_columns

Géométrie des tables

	f_table_catalog character varying (256)	f_table_schema name	f_table_name name	f_geometry_column name	coord_dimension integer	srid integer	type character varying (30)
1	formation	public	route_xy	geom	2	2154	MULTILINESTRING
2	formation	public	commune	geom	2	2154	MULTIPOLYGON
3	formation	public	etablissement	geom	2	2154	POINT
4	formation	public	ponctuel_hydrogra...	geom	2	2154	MULTIPOINT
5	formation	public	troncon_hydrogra...	geom	2	2154	MULTILINESTRING
6	formation	public	iris_extrait72	geom	2	2154	MULTIPOLYGON

L'index spatial

```
CREATE INDEX sidx_commune_geom
ON commune
USING gist
(geom);
```

La vue geography_columns (coordonnées sphériques)

La projection d'une table

geometry_columns

Vue des géométries

Modifier la projection d'une table

SELECT UpdateGeometrySrid ('NomTable','geometry' ,srid);

Reprojeter une table

ST_Transform(geometry,srid)

Effacer la géométrie

SELECT DropGeometryColumn ('NomSchema', 'NomTable','geom');

Effacer une table géométrique

DROP TABLE NomTable;

Exercice

Créer la table *commune_wgs84*, à partir de la table *commune* reprojetée en WGS84

Le schéma, la table et la vue

Le schéma

CREATE SCHEMA nom_du_schema ;
SELECT * FROM nom_schema.nom_table ;

La table

CREATE TABLE nom_table AS [..]
ALTER TABLE, INSERT INTO et DROP

La vue

CREATE VIEW nom_vue AS [..]

Exercice pratique

1. Charger les couches suivantes dans PostGIS comme dans l'exercice du module précédent :

   /BD_TOPO/I_ZONE_ACTIVITE/PAI_SANTE.SHP
   /BD_TOPO/H_ADMINISTRATIF/COMMUNE.SHP
   /BD_TOPO/E_BATI/BATI_INDUSTRIEL.SHP
   /BD_TOPO/F_VEGETATION/ZONE_VEGETATION.SHP

   Créer une nouvelle vue **BATI_INDUSTRIEL10** et la charger dans QGIS en sélectionnant dans la table BATI_INDUSTRIEL les 'Bâtiment industriels' (attention à la majuscule!) dont la hauteur est d'au moins 10 m

2. Créer la vue **conifere** les 'Forêt fermée de conifères' de la commune de La Flèche. Ne pas oubliez de mettre une condition de jointure entre les deux couches... qui devra être ici spatiale.

3. Calculer la somme des surfaces des 'Forêt fermée de feuillus' de la commune de la Flèche en ha (1ha = 10 000 m2), en faisant attention à ne prendre en compte que les parties de surfaces des polygones réellement situées à l'intérieur de la commune.