-- -- ----------------------------------------------------------------

-- -- Le script ci joint traite les données placées dans un schéma nommé ocs\_ge

-- -- -pour le département 35

-- -- les tables ocs\_ge\_2020\_035 et ocsge\_95\_1975 représentent respectivement

-- -- -l’OCS GE du département 35 en 2020 et en 2017 dans le système Lambert93

-- --------------------------------------------------------------------------------------------------

-- Base : ocs\_ge\_ng

-- Schéma : artif\_<dep>

-- Table en entrée : ocsge\_dep\_d1 (ex : ocsge\_95\_1975)

-- Pour un département :

-- 0. remplacer 95 (95 -> 78)

-- 1. exécuter 2 fois ocs\_ge\_artificialisation\_dep\_date.sql - remplacer 2021 (1975 -> 2002)

-- 2. exécuter ocs\_ge\_artificialisation\_dep\_compar.sql

-- --------------------------------------------------------------------------------------------------

------------ EXTRACTION DES DONNEES par dep. et date\_edition ------------

-----------------------------------------------------------------------------------

BEGIN ;

create schema if not exists artif\_<dep> ;

-- depuis ocs\_ge\_ng.metropole.ocsolge, extraire les données du départment "dep", pour la date "date" (40s)

drop table if exists artif\_<dep>.ocsge\_d1;

SELECT \* INTO artif\_<dep>.ocsge\_d1 FROM metropole.ocsolge WHERE insee = '<dep>' AND edition = <date> ;

CREATE INDEX ON artif\_<dep>.ocsge\_d1 USING gist (the\_geom);

CREATE INDEX ON artif\_<dep>.ocsge\_d1 USING btree (cleabs);

------------ PREPARATION DES DONNEES ------------

------------------------------------------------------------

-- nettoyage de la géométrie (20s)

-----------------------------------------

UPDATE artif\_<dep>.ocsge\_d1 set the\_geom = st\_multi(st\_simplify(ST\_Multi(ST\_CollectionExtract(ST\_ForceCollection(ST\_MakeValid(the\_geom)),3)),0))

WHERE st\_geometrytype(the\_geom) in ('ST\_Polygon','ST\_MultiPolygon') and st\_isvalid(the\_geom) is false;

-- ajout d'un champ surface

----------------------------

alter table artif\_<dep>.ocsge\_d1 add column R0 numeric;

UPDATE artif\_<dep>.ocsge\_d1 set R0 = st\_area(the\_geom) ;

-- typage des polygones ocs\_ge en fonction des champs "couvertu" (=code\_cs) et "usage" ( =code\_us)

-- temps : 8s

---------------------------------------------------------------------------------------------------

-- ajout colonne artif/non artif dans ocsge\_d1

alter table artif\_<dep>.ocsge\_d1 add column artif character varying(10);

-- reset : UPDATE artif\_<dep>.ocsge\_d1 set artif=NULL ;

-- artificialisé (bâti + autres)

UPDATE artif\_<dep>.ocsge\_d1

 set artif='artif' WHERE artif is null AND couvertu IN ('CS1.1.1.1') ;

 -- set artif='bati' WHERE artif is null AND couvertu IN ('CS1.1.1.1') ;

UPDATE artif\_<dep>.ocsge\_d1

set artif='artif'

WHERE artif is null

 AND ( couvertu IN ('CS1.1.1.2', 'CS1.1.2.2')

 OR (couvertu = 'CS1.1.2.1' AND NOT usage = 'US1.3')

 OR (couvertu like 'CS2.2%' AND (usage IN ('US2','US3','US5','US235','US6.1','US6.2') OR usage like 'US4%'))) ;

--non artificialisé

UPDATE artif\_<dep>.ocsge\_d1 set artif='non artif' WHERE artif is null;

COMMIT ;

--------------------------------------------------------------------------

-- Fusion des polygones voisins sur valeurs champ "artif"

-- temps : 17 mn

--------------------------------------------------------------------------

BEGIN ;

drop table if exists artif\_<dep>.f0\_d1;

create table artif\_<dep>.f0\_d1 as

SELECT row\_number() over() as gid, artif, st\_unaryunion(unnest(st\_clusterintersecting(the\_geom))) as geom

FROM artif\_<dep>.ocsge\_d1 group by artif;

CREATE INDEX ON artif\_<dep>.f0\_d1 USING gist (geom);

COMMIT ;

BEGIN ;

-- nettoyage des multicollections et calcul des surfaces

-- temps : 6s

drop table if exists artif\_<dep>.z0\_d1;

create table artif\_<dep>.z0\_d1 as

 ( SELECT artif, st\_multi((ST\_Dump(geom)).geom)::geometry(MultiPolygon) as geom from artif\_<dep>.f0\_d1

 WHERE st\_geometrytype(geom) in ('ST\_Polygon','ST\_MultiPolygon'))

 UNION

 ( SELECT artif,st\_multi((ST\_Dump(ST\_CollectionExtract(geom,3))).geom)::geometry(MultiPolygon) as geom from artif\_<dep>.f0\_d1

 WHERE st\_geometrytype(geom) in ('ST\_GeometryCollection'));

ALTER TABLE artif\_<dep>.z0\_d1 ALTER COLUMN geom TYPE geometry(MULTIpolygon, 2154) USING ST\_SetSRID(geom,2154);

CREATE INDEX ON artif\_<dep>.z0\_d1 USING gist (geom);

alter table artif\_<dep>.z0\_d1 add column gid serial not null;

alter table artif\_<dep>.z0\_d1 add column R1 numeric;

UPDATE artif\_<dep>.z0\_d1 set R1 = st\_area(geom);

-- identifier les objets classés artif qui sont dûs au bati pour ne pas y toucher

-- temps : 3mn

drop table if exists artif\_<dep>.bati\_d1 ;

-- bati\_d1 : géométrie MULTIPOINT, union de tous les pointOnSurface du département/date

create table artif\_<dep>.bati\_d1 as (

 WITH bati\_all as ( SELECT st\_pointonsurface(the\_geom)::geometry(Point,2154) as geom

 from artif\_<dep>.ocsge\_d1 WHERE couvertu='CS1.1.1.1')

 SELECT st\_union(geom) as geom from bati\_all) ;

 CREATE INDEX ON artif\_<dep>.bati\_d1 USING gist (geom);

alter table artif\_<dep>.z0\_d1 add column isbati character varying(3);

WITH extr as ( SELECT \* from artif\_<dep>.z0\_d1 WHERE artif='artif' AND R1 < 2500),

 bati\_Ok as ( SELECT gid from extr WHERE st\_intersects ((SELECT geom from artif\_<dep>.bati\_d1), extr.geom))

UPDATE artif\_<dep>.z0\_d1

SET isbati='oui'

WHERE gid in (SELECT gid from bati\_Ok) ;

-- changement de classe des petits objets artifs non bati <2500m2

------------------------------------------------------------------

-- faire une copie de z0\_d1

drop table if exists artif\_<dep>.z1\_d1;

create table artif\_<dep>.z1\_d1 as SELECT \* from artif\_<dep>.z0\_d1;

CREATE INDEX ON artif\_<dep>.z1\_d1 USING gist (geom);

alter table artif\_<dep>.z1\_d1 add column artif0 character varying(10);

UPDATE artif\_<dep>.z1\_d1 set artif0=artif ;

UPDATE artif\_<dep>.z1\_d1 set artif='non artif' WHERE artif='artif' and isbati is null and R1<2500 ;

--------------------------------------------------------------------------------------------------------------------

COMMIT ;

-- Refusionner polygones voisins / artif='non\_artif'

--------------------------------------------------------------------------

BEGIN ;

drop table if exists artif\_<dep>.f1\_d1 ;

create table artif\_<dep>.f1\_d1 as

 ( SELECT artif, st\_multi(st\_unaryunion(unnest(st\_clusterintersecting(geom)))) as geom FROM artif\_<dep>.z1\_d1

 WHERE artif='non artif' group by artif )

 UNION

 ( SELECT artif, st\_multi(geom) from artif\_<dep>.z1\_d1 WHERE artif='artif' ) ;

ALTER TABLE artif\_<dep>.f1\_d1 ALTER COLUMN geom TYPE geometry(MULTIpolygon, 2154) USING ST\_SetSRID(geom,2154);

CREATE INDEX ON artif\_<dep>.f1\_d1 USING gist (geom);

-- nettoyage des multicollections et calcul des surfaces

drop table if exists artif\_<dep>.z2\_d1;

create table artif\_<dep>.z2\_d1 as

 ( SELECT artif,st\_multi((ST\_Dump(geom)).geom)::geometry(MultiPolygon,2154) as geom from artif\_<dep>.f1\_d1

 WHERE st\_geometrytype(geom) in ('ST\_Polygon','ST\_MultiPolygon'))

 UNION

 ( SELECT artif,st\_multi((ST\_Dump(ST\_CollectionExtract(geom,3))).geom)::geometry(MultiPolygon,2154) as geom from artif\_<dep>.f1\_d1

 WHERE st\_geometrytype(geom) in ('ST\_GeometryCollection'));

CREATE INDEX ON artif\_<dep>.z2\_d1 USING gist (geom);

alter table artif\_<dep>.z2\_d1 add column R2 numeric ;

UPDATE artif\_<dep>.z2\_d1 set R2 = st\_area(geom);

--------------------------------------------------------------

-- changement de classe des petits objets non artif <2500m2

--------------------------------------------------------------

alter table artif\_<dep>.z2\_d1 add column artif0 character varying(10);

UPDATE artif\_<dep>.z2\_d1 set artif0=artif ;

UPDATE artif\_<dep>.z2\_d1 set artif='artif' WHERE artif='non artif' and R2<2500;

COMMIT ;

-------------------------------------------------------------------------

-- table artif\_<dep>.artif\_ocsge\_d1 : polygones OCSGE initiaux

-- Nouveau champ "impact\_seuil" booléen

-- impact\_seuil = true si le seuil de 2500m2 a inversé la valeur initiale de "artif"

-- Durée du traitement : 2 à 6 mn...

-------------------------------------------------------------------------

BEGIN ;

DROP table if exists artif\_<dep>.artif\_ocsge\_d1 ;

CREATE table artif\_<dep>.artif\_ocsge\_d1 as (SELECT \* from artif\_<dep>.ocsge\_d1) ;

CREATE INDEX ON artif\_<dep>.artif\_ocsge\_d1 USING gist (the\_geom);

ALTER table artif\_<dep>.artif\_ocsge\_d1 add column impact\_seuil boolean default(false);

-- inversion 'artif'-> 'non artif'

-- -------------------------------

WITH inv as (

 select st\_union(geom) as geom from artif\_<dep>.z1\_d1

 where artif0 = 'artif' and artif = 'non artif'),

 tri as (

 select a.gid, st\_pointonsurface(the\_geom)::geometry(Point,2154) as tri\_geom

 from artif\_<dep>.ocsge\_d1 a, inv

 where a.artif = 'artif' and R0<2500 and st\_intersects(geom, the\_geom))

UPDATE artif\_<dep>.artif\_ocsge\_d1 a SET impact\_seuil = true, artif = 'non artif'

FROM tri

WHERE a.gid=tri.gid AND st\_intersects(tri\_geom, the\_geom) ;

-- inversion 'non artif'-> 'artif'

-- -------------------------------

WITH inv as (

 select st\_union(geom) as geom from artif\_<dep>.z2\_d1

 where artif0 = 'non artif' and artif = 'artif'),

 tri as (

 select a.gid, st\_pointonsurface(the\_geom)::geometry(Point,2154) as tri\_geom

 from artif\_<dep>.ocsge\_d1 a, inv

 where a.artif = 'non artif' and R0<2500 and st\_intersects(geom, the\_geom))

UPDATE artif\_<dep>.artif\_ocsge\_d1 a SET impact\_seuil = true, artif = 'artif'

FROM tri

WHERE a.gid=tri.gid AND st\_intersects(tri\_geom, the\_geom) ;

UPDATE artif\_<dep>.artif\_ocsge\_d1 set R0 = round(cast(R0 as numeric),2);

COMMIT ;