

Requête SQL utiles

Correspondance entre code INSEE présent dans additional_data et cor_area_synthese

Trouver les codes INSEE fournis dans le champ additional_data attribut communeInseeCode, existant dans la table ref_geo.l_areas mais qui ne correspondent pas à ceux présent dans la table gn_synthese.cor_area_synthese :

```
WITH communes AS (  
    SELECT la.id_area, la.area_code AS insee_code, la.area_name  
    FROM ref_geo.l_areas AS la  
    WHERE la.id_type = ref_geo.get_id_area_type_by_code('COM')  
    AND la."enable" = TRUE  
)  
SELECT s.unique_id_sinp, s.the_geom_4326,  
s.additional_data::json->>'communeInseeCode' AS code_insee_json, c.area_name  
AS area_name_cas, c.insee_code AS code_insee_cas  
FROM gn_synthese.synthese AS s  
    LEFT JOIN gn_synthese.cor_area_synthese AS cas  
        ON (s.id_synthese = cas.id_synthese)  
    JOIN communes AS c  
        ON (cas.id_area = c.id_area)  
WHERE s."precision" IS NULL  
    AND s.additional_data::json->>'communeInseeCode' != c.insee_code ;
```

Trouver les codes INSEE fournis dans le champ additional_data attribut communeInseeCode qui ne correspondent pas à ceux présent dans la table gn_synthese.cor_area_synthese car ils n'existent pas dans la table ref_geo.l_areas :

```
WITH communes AS (  
    SELECT la.id_area, la.area_code AS insee_code, la.area_name  
    FROM ref_geo.l_areas AS la  
    WHERE la.id_type = ref_geo.get_id_area_type_by_code('COM')  
    AND la."enable" = TRUE  
)  
SELECT DISTINCT s.additional_data::json->>'communeInseeCode' AS  
code_insee_json  
FROM gn_synthese.synthese AS s  
    LEFT JOIN gn_synthese.cor_area_synthese AS cas  
        ON (s.id_synthese = cas.id_synthese)  
    JOIN communes AS c  
        ON (cas.id_area = c.id_area)  
WHERE s."precision" IS NULL  
    AND s.additional_data::json->>'communeInseeCode' != c.insee_code  
    AND s.additional_data::json->>'communeInseeCode' NOT IN (SELECT  
insee_code FROM communes);
```

Calculer le rayon du cercle comprenant un polygone (communes)

```
SELECT
    unique_id_sinp,
    round(radius(ST_MinimumBoundingRadius(la.geom))) AS "precision",
    center(ST_MinimumBoundingRadius(la.geom)) AS rayon,
    ST_MinimumBoundingCircle(la.geom) AS cercle,
    ST_LongestLine(center(ST_MinimumBoundingRadius(la.geom)),
ST_MinimumBoundingCircle(la.geom)) AS rayon,
    st_centroid(la.geom) AS centroid,
    la.geom,
    la.area_name
FROM gn_synthese.synthese AS s
    LEFT JOIN gn_synthese.cor_area_synthese AS cas
        ON (s.id_synthese = cas.id_synthese)
    JOIN ref_geo.l_areas AS la
        ON (cas.id_area = la.id_area)
WHERE s.id_source != gn_synthese.get_id_source_by_name('SI CBN')
    AND s."precision" IS NULL
    AND la.id_type = ref_geo.get_id_area_type_by_code('COM')
LIMIT 100;
```

From:

<https://sinp-wiki.cbn-alpin.fr/> - **CBNA SINP**

Permanent link:

<https://sinp-wiki.cbn-alpin.fr/database/requetes-sql-utiles?rev=1634134750>

Last update: **2021/10/13 14:19**

