

Exercice 7 et sa solution

QGIS 3.34 Septembre 2024



QGIS Perfectionnement

Table des matières

I - Exercice : Exercice 7 : Requêtes et fonctions spatiales	3
Solutions des exercices	5

Exercice : Exercice 7 : Requêtes et fonctions spatiales



Réaliser des requêtes et fonctions spatiales

Objectif : En utilisant les tables de 'QGIS_perf_sandbox.sqlite' réaliser les 4 requêtes suivantes

Question 1

[solution n°1 p. 5]

Q1 : quels sont les ponctuels hydrographiques de la commune de La Flèche ?

Indice :

On utilisera les tables **PONCTUEL_HYDROGRAPHIQUE** et **COMMUNE...** trouver la relation géométrique entre **PONCTUEL_HYDROGRAPHIQUE.Geometry** et **COMMUNE.Geometry**.

Question 2

[solution n°2 p. 5]

Q2 : quelle est la longueur de la 'rivière le loir' dans chacune des communes intersectées par le cours d'eau ?

résultat à obtenir :

	NOM_COMM	TOPONYME	longueur
1	BAZOUGES-SUR-LE-LOIR	rivière le loir	2251.34
2	CLERMONT-CREANS	rivière le loir	1351.55
3	CRE	rivière le loir	1034.7
4	LA FLECHE	rivière le loir	14155.64
5	MAREIL-SUR-LOIR	rivière le loir	1079.16
6	THOREE-LES-PINS	rivière le loir	283.56

résultat exo7 question 2

Indice :

Chaque commune peut contenir plusieurs tronçons, il faut donc calculer la somme des longueurs des tronçons pour chaque commune... donc utiliser un **GROUP BY** et une fonction **sum()**.

On utilisera la fonction **ST_Length()** pour obtenir la longueur de chaque tronçon. Il faut également tenir compte que certains tronçons sont à cheval sur plusieurs communes, et donc ne prendre en compte que la longueur des tronçons qui sont à l'intérieur de chaque commune.

Pour ce faire on utilisera **st_intersection(a.geom, b.geom)** qui permet de récupérer la géométrie de l'objet a qui intersecte celle de l'objet b.

Question 3

[solution n°3 p. 6]

Q3 : sélectionner les 'ponctuels hydrographiques' qui sont à moins de 5 km d'un établissement d'enseignement (couche ETABLISSEMENT)

Indice :

On pourra utiliser une fonction **st_distance()** ou une fonction **st_buffer()** associée à un opérateur de type **st_contains()** ou **st_intersects()**.

Question 4

[solution n°4 p. 6]

Q4 : Quel est l'établissement le plus proche du centroïde de la commune de la Flèche?

On utilisera les coordonnées X_COMMUNE et Y_COMMUNE et la fonction **st_makepoint()** ou **Makepoint()** sous spatialite ou encore **st_point()** qui est un alias de **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 **srid()** qui récupère le srid d'une géométrie.

Indice :

utiliser la fonction **distance()**, **ORDER BY** pour trier et **LIMIT 1** pour prendre le 1er objet renvoyé.

Une requête de type

```
SELECT nom_comm, srid(Geometry) AS SRID, MakePoint(X_COMMUNE, Y_COMMUNE,
srid(Geometry)) AS Geometry FROM commune WHERE commune.nom_comm = 'LA
FLECHE'
```

retourne des points au centroïde calculé à partir des coordonnées X_COMMUNE, Y_COMMUNE.

Solutions des exercices



[exercice p. 3] Solution n°1

Solution :

```
SELECT * FROM ponctuel_hydrographique, commune WHERE  
st_within(ponctuel_hydrographique.Geometry, commune.Geometry) AND  
commune.nom_comm = 'LA FLECHE'
```

On retient donc les ponctuels qui sont dans (st_within) les communes et on sélectionne uniquement la commune de la Flèche avec AND commune.nom_comm = 'LA FLECHE'

Il doit y avoir 81 ponctuels dans le résultat.

[exercice p. 3] Solution n°2

```
SELECT nom_comm, toponyme,  
round(sum(st_length(st_intersection(troncon_hydrographique.Geometry,  
commune.Geometry))),2) as longueur FROM troncon_hydrographique,  
commune WHERE toponyme= 'rivière le loir' AND  
st_intersects(commune.geometry, troncon_hydrographique.geometry) GROUP  
BY nom_comm
```

Pour information : sous PostGIS

```
SELECT nom_comm, toponyme,  
round(sum(st_length(st_intersection(troncon_hydrographique.Geometry,  
commune.Geometry))) : : numeric,2) as longueur FROM  
troncon_hydrographique, commune WHERE toponyme= 'rivière le loir' AND  
st_intersects(commune.geometry, troncon_hydrographique.geometry) GROUP  
BY nom_comm
```

Le résultat est le suivant :

Résultat :			
	NOM_COMM	TOPONYME	longueur
1	BAZOUGES-SUR-LE-LOIR	rivière le loir	2251.34
2	CLERMONT-CREANS	rivière le loir	1351.55
3	CRE	rivière le loir	1034.7
4	LA FLECHE	rivière le loir	14155.64
5	MAREIL-SUR-LOIR	rivière le loir	1079.16
6	THOREE-LES-PINS	rivière le loir	283.56

résultat Exo7 Question 2

Lorsque les requêtes sont un peu longues, il peut-être utile d'**utiliser des alias pour les noms des tables**. Ils s'obtiennent en plaçant directement un alias après le nom de table dans la clause FROM

Dans le cas précédent cela donne :

```
SELECT      nom_comm,      toponyme,
round(sum(st_length(st_intersection(a.Geometry,b.Geometry))),2)      as
longueur FROM troncon_hydrographique a, commune b WHERE toponyme=
'rivière le loir' AND st_intersects(a.geometry, b.geometry) GROUP BY
nom_comm
```

[exercice p. 4] **Solution n°3**

```
SELECT * FROM ponctuel_hydrographique, etablisement WHERE
st_distance(ponctuel_hydrographique.Geometry, etablisement .Geometry)
< 5000
```

ou

```
SELECT * FROM ponctuel_hydrographique, etablisement WHERE
st_contains(Buffer(etablisement.Geometry, 5000),
ponctuel_hydrographique.Geometry)
```

Pour information sous PostGIS on pourrait utiliser la fonction st_Dwithin()

```
SELECT * from ponctuel_hydrographique, etablisement WHERE
ST_DWithin(ponctuel_hydrographique.Geometry, etablisement .Geometry,
5000)
```

On remarquera cependant qu'avec ces requêtes les ponctuels qui sont à moins de 5000 m de plusieurs établissements apparaissent plusieurs fois.

On peut ajouter une clause **GROUP BY pkuid** pour ne les obtenir qu'une seule fois (dans ce cas on n' a pas tous les établissements situés à moins de 5000 m d'un ponctuel hydrographique, mais ce n'est pas la question).

Il doit y avoir 86 ponctuels avec le group by et 150 réponses **sans le group by**

Une autre solution est d'utiliser une union des buffers individuels autour de chaque établissement avec la fonction st_union().

La syntaxe qui utilise une sous-requête directement dans la fonction st_contains() est alors :

```
SELECT * FROM PONCTUEL_HYDROGRAPHIQUE WHERE ST_contains((SELECT
st_union(st_buffer(Geometry,5000)) FROM ETABLISSEMENT)
,PONCTUEL_HYDROGRAPHIQUE.Geometry)
```

[exercice p. 4] **Solution n°4**

```
SELECT *, st_distance( MakePoint(X_COMMUNE, Y_COMMUNE,
srid(COMMUNE.Geometry)), etablisement .Geometry) AS distance FROM
etablisement , COMMUNE WHERE commune.nom_comm = 'LA FLECHE' ORDER BY
distance LIMIT 1
```

On aurait pu également utiliser la fonction st_centroid() qui donne un centroïde calculé, mais qui est différent des coordonnées données dans X_COMMUNE et Y_COMMUNE.

L'identifiant BdCARTO de l'Établissement répondant à la question est 123049.