

Jour 14 - Requêtes topologiques

Les données OpenStreetMap sont fortement **topologiques**. Les routes sont **connectées** par des nodes pour former un graphe, il en est de même pour les rivières, les voies ferrées, les infrastructures électriques. Les limites administratives adjacentes **partagent** des ways, les bâtiments adjacents partagent des nodes, etc.

La **réursion** permet de « naviguer » au sein de cette topologie. Commençons par le **partage des nodes** entre les routes. La requête suivante trouve les rues croisant la Rue de Belleville à Paris :

```
area["ref:INSEE"]=75056];
way[highway][name="Rue de Belleville"](area);
node(w);
way(bn)[highway];
out geom;
```

Les instructions exploitant la topologie sont :

1. `node(w)` récupère les nodes référencés par les ways qui constituent la Rue de Belleville;
2. `way(bn)` trouve les ways qui référencent ces nodes, le filtre `[highway]` ne conservant que les rues.

Cette requête ne récupère que les ways *directement connectés* à la rue de Belleville : la Rue des Pyrénées n'est par exemple pas complète. Pour cela il convient de remonter aux relations définissant l'ensemble des rues connectées, puis d'en récupérer les ways (ce qui permet au passage de laisser de côté les adresses) :

```
area["ref:INSEE"]=75056];
way[highway][name="Rue de Belleville"](area);
node(w);
way(bn)[highway];
rel(bw)[type=associatedStreet];
way(r);
out geom;
```

Passons au **partage de ways**. La requête suivante utilise la récursion montante des ways vers les relations, pour trouver les communes dont une partie du contour est définie par la rivière La Risle, en Normandie :

```
area[admin_level=4][name="Normandie"];
way[waterway=river][name="La Risle"](area);
out geom;
rel(bw)[boundary=administrative][admin_level=8];
out geom;
```

Le résultat de cette requête topologique est à comparer à la requête géométrique vue le jour 11, qui récupérait les communes traversées par l'autoroute A87. Ici seules les communes dont le contour *partage un way* avec la rivière sont retournées.

Ce partage de nodes ou de ways est l'occasion d'introduire **l'intersection** entre lots de données - nous avons déjà vu **l'union** avec (`.lot1; .lot2;`); et **la soustraction** avec (`.lot1; - .lot2;`);. L'intersection de lots de données nécessite d'en avoir au moins deux, et donc d'utiliser des variables. La syntaxe est `<type>.lot1.lot2;` où `<type>` peut être `node`, `way`, `rel` ou même `nwr` : le résultat contient les éléments qui sont à la fois dans `.lot1` **et** dans `.lot2`.

Restons en Normandie pour trouver la limite entre deux communes riveraines :

```
area[admin_level=4][name="Normandie"]->.a;
rel[admin_level=8][name="Manneville-sur-Risle"](area.a);
way(r)->.contour1;
rel[admin_level=8][name="Pont-Audemer"](area.a);
way(r)->.contour2;
way.contour1.contour2;
out geom;
```

Il est en fait possible de faire l'intersection entre plus de deux lots de données, avec la syntaxe `<type>.lot1.lot2.lot3;` : un exemple d'usage est en exercice...

Exercices

- Trouvez les routes primaires ou secondaires croisant l'itinéraire cyclable "La Coulée Verte du Sud Parisien".
- Trouvez les intersections entre les routes "primaires" et les pistes cyclables dans l'Essonne.
- Trouvez les communes voisines de votre commune.
- Trouvez les bâtiments de Paris (ou une autre commune) sur le mur desquels est fixé un distributeur de billets.
- Trouvez la section de la rivière L'Erdre séparant les communes de Nantes et La-Chapelle-sur-Erdre.

© CC-by-sa Carto'Cité

From:

<http://wiki.cartocite.fr/> -



Permanent link:

http://wiki.cartocite.fr/doku.php?id=tutoverpass:jour_14_requetes_topologiques&rev=1597829460

Last update: 2021/01/25 17:12

