

Jour 8 - Utiliser des variables

Cherchons maintenant les parcs se trouvant à Saint-Denis **et** à La Courneuve. Si nous appliquons la requête suivante, le Parc départemental Georges-Valbon, qui se trouve à cheval sur les 2 communes, est retourné 2 fois.

```
area["ref:INSEE"="93066"]; // Saint-Denis
nwr[leisure=park](area);
out geom;
area["ref:INSEE"="93027"]; // La Courneuve
nwr[leisure=park](area);
out geom;
```

Pour éviter cela nous avons vu le jour 4 comment produire l'union de plusieurs lots de données. La requête suivante fonctionne, mais elle a un problème : savez-vous le trouver ?

```
(
  area["ref:INSEE"="93066"]; // Saint-Denis
  nwr[leisure=park](area);
  area["ref:INSEE"="93027"]; // La Courneuve
  nwr[leisure=park](area);
);
out geom;
```

Le problème est qu'elle inclut dans le résultat les deux surfaces Saint-Denis et La Courneuve, car les instructions `area` se trouvent à l'intérieur de l'union. Certes leur géométrie n'est pas retournée, car l'instruction `area` ne le permet pas, mais nous ne les souhaitons pas dans le résultat de la requête.

Pour résoudre ce problème nous devons utiliser des **variables**. Une variable est un mot arbitraire qui permet de stocker un lot de données, afin de le réutiliser à tout endroit dans la suite de la requête. La solution consiste donc à placer chaque surface dans une variable, que nous utilisons pour trouver les parcs de chaque commune :

```
area["ref:INSEE"="93066"]->.sd; // Saint-Denis
area["ref:INSEE"="93027"]->.lc; // La Courneuve
(
  nwr[leisure=park](area.sd); // Parcs de Saint-Denis
  nwr[leisure=park](area.lc); // Parcs de La Courneuve
);
out geom;
```

Notez bien ces éléments de syntaxe :

1. Une variable est un mot arbitraire commençant par un point, ici `.sd` et `.lc`
2. Stocker un lot de données dans un variable s'écrit avec les signes `->` (qui dessinent une flèche)
3. `(area.sd)` est un filtre permettant de sélectionner les éléments à l'intérieur **des** surfaces contenues dans le lot de données `.sd`

Il existe une autre façon d'écrire la même requête. Elle consiste à produire un lot de données

contenant les deux surfaces qui nous intéressent, avec une **union**, pour ensuite chercher les parcs se trouvant dans l'une ou l'autre :

```
(
  area["ref:INSEE"="93066"]; // Saint-Denis
  area["ref:INSEE"="93027"]; // La Courneuve
)->.villes;
nwr[leisure=park](area.villes);
out geom;
```

On peut même alors se passer de la variable...

```
(
  area["ref:INSEE"="93066"]; // Saint-Denis
  area["ref:INSEE"="93027"]; // La Courneuve
);
nwr[leisure=park](area);
out geom;
```

Vous savez quoi ? Ce tuto me fait réaliser que le filtre (area) sélectionne en réalité les éléments se trouvant **au moins en partie** à l'intérieur d'une surface, pas seulement ceux se trouvant **entièrement** dedans. En termes d'analyse spatiale, on obtient donc les éléments *intersectant* une surface.

Exercices

- Trouvez les chemins et pistes cyclables se trouvant dans le Parc départemental Georges-Valbon
- Trouvez les hôtels et restaurants de Saint-Denis et La Courneuve

© CC-by-sa Carto'Cité

From:

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

Carto^oCité

Permanent link:

http://wiki.cartocite.fr/doku.php?id=tutoverpass:jour_8_utiliser_des_variables

Last update: **2021/01/25 17:14**

