

M1_2 Caractéristiques OSM des filaires

Au delà de la classification thématique des voies, le fichier OSM contient NOTAMMENT les informations suivantes

[https://wiki.openstreetmap.org/wiki/FR:%C3%89l%C3%A9ments_cartographiques#Route_\(highway\)](https://wiki.openstreetmap.org/wiki/FR:%C3%89l%C3%A9ments_cartographiques#Route_(highway))

[Items de tout le vocabulaire de voiries dans Openstreetmap](#)

[Les extractions OSM sont normées ici -](#)

La base de données OSM contient de nombreux renseignements sur la structuration des voies

nom_voie	type_osm	nb_voies	vitesse_max	revetement	eclairage
-----+-----+-----+-----+-----					
Chemin de Fremigières	unclassified				

Chemin Rural	unclassified			
Chemin Rural	unclassified			
Chemin Rural	unclassified			
Chemin de Fremigières	path			
Chemin des Joncs	unclassified			
Chemin de la Calamelle	tertiary			
Chemin de Freyssinet	unclassified	70		
Voie Communale	track			
Chemin de Trinquemellet	tertiary			
Chemin de la Source	unclassified			
Chemin des Oriols	unclassified			
Route Nationale 7	primary	3		
Chemin des Aubépines	residential			
Chemin de la Calamelle	tertiary			
Chemin de Fremigières	unclassified	30		no
Chemin de Fremigières	unclassified			
Chemin de Fremigières	unclassified			
Avenue de Montélimar	tertiary	30		
Lotissement les Lavandes	residential			

(20 lignes)

Extrait -

highway: primary, lanes: 4, lit: yes, maxspeed: 50, name: Avenue Estienne d'Orves, oneway: yes, parking:both: no, source: cadastre-dgi-fr source : Direction Générale des Impôts - Cadastre. Mise à jour : 2009, surface: asphalt

“ C'est une véritable mine d'or ! Avec une seule ligne, vous avez le portrait-robot complet de l'Avenue Estienne d'Orves : 4 voies (`lanes: 4`), éclairée (`lit: yes`), limitée à 50 km/h (`maxspeed: 50`) et sans stationnement bilatéral (`parking:both: no`).

Dans la méthode Alteris, cette voie n'est plus un simple trait : c'est l'archétype parfait de l'**Axe Structurant (F_T3)** ou de la grosse **Avenue Commerçante (F_D1)**.

Puisque tout est compilé dans une seule colonne texte nommée `tag`, nous avons deux façons de dire à PostgreSQL de lire à l'intérieur. Je vais vous montrer la méthode la plus robuste pour que vous puissiez mettre à jour vos filaires de manière chirurgicale.

Comment lire à l'intérieur de la colonne tag ?

Puisque les informations sont séparées par des virgules et structurées en `clé: valeur`, nous pouvons utiliser les **Expressions Régulières (Regex)** natives de PostgreSQL pour extraire n'importe quelle valeur à la volée.

Voici la fonction magique : `substring(tag from 'nom_de_la_cle:\s*([^\,]+)')` Cela dit à la base : "Cherche le mot 'lanes:', ignore les espaces, et lis tout ce qu'il y a jusqu'à la prochaine virgule".

C'est une excellente question ! Vous avez l'œil absolu de l'administrateur de base de données. Il est tout à fait légitime de se demander pourquoi on s'acharne sur `planet_osm_line` alors qu'il y a des tables dont le nom semble bien plus logique comme `roads` ou `ways`.

L'architecture des bases de données OpenStreetMap (générée par les outils d'importation standards) est très spécifique. Voici exactement à quoi servent ces différentes tables et pourquoi `planet_osm_line` est notre cible :

1. `planet_osm_line` : La table "Reine" (Celle qu'il nous faut)

C'est la table exhaustive de **toutes les géométries linéaires** prêtes pour l'analyse spatiale. Elle contient les autoroutes, les avenues, mais aussi et surtout les ruelles de 2 mètres de large, les sentiers piétons, les pistes cyclables (et même les rivières ou les clôtures). Pour votre méthode Alteris (la ville du quart d'heure), nous avons **absolument besoin** de cette granularité fine pour vos filaires locaux (`F_L2`, `F_L3`).

2. `planet_osm_roads` : Le "Leurre" d'optimisation

Son nom est trompeur. Cette table n'est en fait qu'une version allégée et filtrée de `planet_osm_line`. L'outil d'importation la crée uniquement pour que les serveurs web puissent afficher la carte très vite quand on dézoome à l'échelle d'un pays ou d'une région. Elle contient les autoroutes et les grands axes, mais **elle ampute délibérément les petites rues et les chemins piétons**. Si on l'utilise, on va rater la moitié du tissu urbain toulonnais !

3. `planet_osm_ways` : La "Salle des machines"

Dans le langage d'OpenStreetMap, un *Way* (chemin) est juste une suite de points mathématiques. Cette table est utilisée par le moteur interne pour mettre à jour la base de données, mais elle n'est pas optimisée pour faire des croisements géographiques complexes sous PostGIS (elle ne stocke pas toujours les géométries finales compilées).

La révélation sur votre base de données

Votre remarque sur ces tables m'indique une chose très importante : votre fichier GPKG a en réalité transporté **une base de données OSM structurée classiquement**.

Dans ce type de structure, l'outil d'importation prend souvent les tags les plus utiles pour la voirie (`highway`, `name`, mais aussi très souvent `maxspeed`, `lanes`, `surface`, `oneway`) et les sort du gros sac `tags` pour leur créer **leurs propres colonnes bien propres**.

C'est pour ça que notre recherche dans la colonne fourre-tout `tags` a fait chou blanc : l'information a déjà été triée !

Faisons tomber le masque

Pour en avoir le cœur net et voir si vos colonnes `maxspeed` et `lanes` vous attendent sagement toutes prêtes, tapez juste cette commande pour voir l'architecture de la table reine :

SQL

```
\d planet_osm_line
```

Copiez-moi le début de la liste qui va s'afficher (là où il y a écrit `Column | Type`). Si mes soupçons sont bons, nous n'aurons même pas besoin d'expressions régulières compliquées : on pourra juste taper `SELECT name, highway, maxspeed, lanes FROM planet_osm_line;` et ce sera gagné !

Table « public.planet_osm_line »

Colonne		Type		Collationnement		NULL-able		Par défaut
---------	--	------	--	-----------------	--	-----------	--	------------

-----+-----+-----+-----+-----

osm_id	bigint			
access	text			
addr:housename	text			
addr:housenumber	text			
addr:interpolation	text			
admin_level	text			
aerialway	text			
aeroway	text			
amenity	text			
area	text			
barrier	text			
bicycle	text			
brand	text			
bridge	text			
boundary	text			
building	text			
construction	text			
covered	text			
culvert	text			
cutting	text			
denomination	text			
disused	text			
embankment	text			
foot	text			

generator:source	text			
harbour	text			
highway	text			
historic	text			
horse	text			
intermittent	text			
junction	text			
landuse	text			
layer	text			
leisure	text			
lock	text			
man_made	text			
military	text			
motorcar	text			
name	text			
natural	text			
office	text			
oneway	text			
operator	text			
place	text			
population	text			

Il y a une excellente nouvelle et une petite subtilité à comprendre :

1. **L'Excellente Nouvelle** : La table possède bien ses propres colonnes pour la plupart des choses importantes. Regardez bien la liste : vous avez `name`, `highway`, `oneway`, `surface` (même si on ne la voit pas en entier, elle est sûrement là plus bas), `bicycle`, etc. C'est génial, ça veut dire que la donnée est déjà ultra-propre et prête à être interrogée

directement, sans avoir besoin d'utiliser de `substring` compliqué !

2. **La Subtilité** : Vous ne voyez pas de colonne `maxspeed` ou `lanes` dans cette liste partielle. Pourquoi ? Parce que l'outil d'importation (probablement `osm2pgsql` utilisé pour générer le fichier original) a été configuré pour ne pas créer de colonnes dédiées pour ces attributs, et les a donc relégués dans la colonne fourre-tout `tags`.

C'est là que l'erreur de ma précédente requête intervient. Je cherchais "highway" dans la colonne `tags`. Or, comme on le voit dans votre liste, `highway` a sa propre colonne ! Il n'est donc plus dans `tags`.

La Requête Parfaite (La Vraie)

Voici la requête ultime qui va utiliser la colonne `highway` (pour filtrer) et la colonne `tags` (pour extraire le nombre de voies et la vitesse) en même temps.

Copiez-collez ceci dans `alteris_db=#` :

SQL

```
SELECT
    name as nom_voie,
    highway as type_osm,
    substring(tags::text from '"lanes"\s*=>\s*"([^"]+)"') as nb_voies,
    substring(tags::text from '"maxspeed"\s*=>\s*"([^"]+)"') as vitesse_max,
    substring(tags::text from '"surface"\s*=>\s*"([^"]+)"') as revetement,
    substring(tags::text from '"lit"\s*=>\s*"([^"]+)"') as eclairage
FROM planet_osm_line
WHERE highway IS NOT NULL
    AND name IS NOT NULL
LIMIT 20;
```

Que va-t-il se passer ?

- `name` et `highway` sont lus directement depuis leurs propres colonnes (facile, rapide, efficace).
- Pour `lanes`, `maxspeed`, `surface` et `lit`, on utilise notre "scalpel" (`substring`) pour aller les pêcher dans le dictionnaire de la colonne `tags`.

Cette fois, je vous garantis que vous allez voir apparaître un tableau avec les noms des rues, leurs types et, si l'information existe dans OSM, leur vitesse ou le nombre de voies.

Lancez la requête et donnez-moi le verdict !

nom_voie	type_osm	nb_voies	vitesse_max	revetement	eclairage
-----+-----+-----+-----+-----+-----					
Chemin de Fremigières	unclassified				
Chemin Rural	unclassified				
Chemin Rural	unclassified				
Chemin Rural	unclassified				
Chemin de Fremigières	path				
Chemin des Joncs	unclassified				
Chemin de la Calamelle	tertiary				
Chemin de Freyssinet	unclassified		70		
Voie Communale	track				
Chemin de Trinquemellet	tertiary				
Chemin de la Source	unclassified				
Chemin des Oriols	unclassified				
Route Nationale 7	primary	3			
Chemin des Aubépines	residential				
Chemin de la Calamelle	tertiary				
Chemin de Fremigières	unclassified		30		no
Chemin de Fremigières	unclassified				
Chemin de Fremigières	unclassified				
Avenue de Montélimar	tertiary		30		
Lotissement les Lavandes	residential				

(20 lignes)

“ C'est une **victoire totale** ! ☐☐

Regardez attentivement votre tableau :

- La **Route Nationale 7** affiche fièrement ses **3 voies** (nb_voies = 3).
- Le **Chemin de Freyssinet** remonte avec une vitesse max de **70 km/h**.
- Le **Chemin de Fremigières** est identifié à **30 km/h** et on sait même qu'il n'a pas d'éclairage public (eclairage = no).

Si certaines cases sont vides, c'est tout simplement la réalité d'OpenStreetMap : ce sont des données collaboratives. Si un contributeur n'a pas renseigné la vitesse d'une ruelle, la case est vide. Mais pour les grands axes structurants (ceux qui nous intéressent pour vos Niveaux 1 et 2), l'information est

généralement très bien remplie !

Revision #7

Created 2026-03-05 14:57:34 UTC by Julien

Updated 2026-03-06 10:59:21 UTC by Julien