

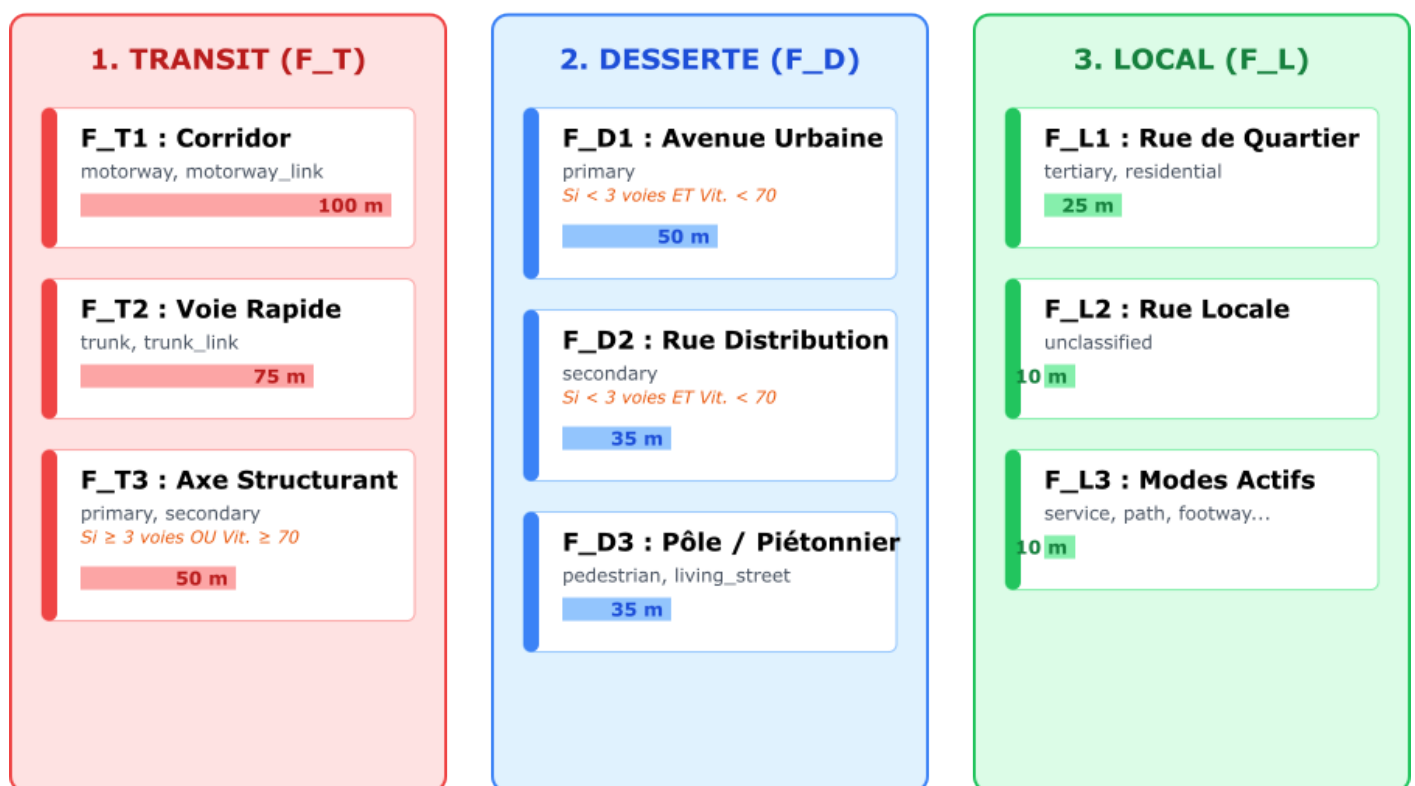
M1_les filaires (matrice des voies)

- [M1_1 - La hiérarchisation des filaires \(axes, routes et voies\)](#)
- [M1_2 Caractéristiques OSM des filaires](#)

M1_1 - La hiérarchisation des filaires (axes, routes et voies)

Modèle de Hiérarchisation Viaire - Méthode ALTERIS

Classification OSM et Emprises Spatiales (Buffers)



La largeur de la barre de couleur indique l'ampleur du rayon d'action spatial (buffer) appliqué par Alteris.

1- La Grille des filaires

La base des filaires - la grille - c'est la classification OSM-expert

[Le tableau est ici](#)

```

DROP TABLE IF EXISTS zones_motifs_filaires_v3 CASCADE;

CREATE TABLE zones_motifs_filaires_v3 AS
WITH extraction AS (
    -- Étape 1 : On lit la géométrie et on extrait les informations clés du JSON
    SELECT
        osm_id,
        tags->>'name' AS nom_voie,
        tags->>'highway' AS type_osm,
        tags->>'lanes' AS nb_voies,
        tags->>'maxspeed' AS vitesse_max,
        ST_Transform(geom, 2154) AS geom_ligne -- Projection en Lambert 93 (mètres)
    FROM osm_lignes
    WHERE tags->>'highway' IS NOT NULL
),
classification AS (
    -- Étape 2 : On applique VOS règles d'urbanisme (L'entonnoir Alteris)
    SELECT
        osm_id, nom_voie, type_osm, geom_ligne,
        CASE
            -- 1. TRANSIT (F_T)
            WHEN type_osm IN ('motorway', 'motorway_link') THEN 'F_T1'
            WHEN type_osm IN ('trunk', 'trunk_link') THEN 'F_T2'
            WHEN type_osm IN ('primary', 'secondary')
                AND (nb_voies IN ('3', '4', '5', '6', '7', '8')
                    OR vitesse_max IN ('70', '80', '90', '110', '130'))
            THEN 'F_T3'

            -- 2. DESSERTE (F_D)
            -- Note : Si une route 'primary' arrive ici, c'est qu'elle a échoué au test F_T3
            WHEN type_osm = 'primary' THEN 'F_D1'
            WHEN type_osm = 'secondary' THEN 'F_D2'
            WHEN type_osm IN ('pedestrian', 'living_street') THEN 'F_D3'

            -- 3. LOCAL (F_L)
            WHEN type_osm IN ('tertiary', 'residential') THEN 'F_L1'
            WHEN type_osm = 'unclassified' THEN 'F_L2'
            ELSE 'F_L3'
        END as code_pattern

```

```

FROM extraction
)
-- Étape 3 : On génère les emprises spatiales (buffers) avec les bonnes largeurs
SELECT
    osm_id,
    nom_voie,
    code_pattern,
    ST_Buffer(geom_ligne,
        CASE
            WHEN code_pattern = 'F_T1' THEN 100
            WHEN code_pattern = 'F_T2' THEN 75
            WHEN code_pattern = 'F_T3' THEN 50
            WHEN code_pattern = 'F_D1' THEN 50
            WHEN code_pattern = 'F_D2' THEN 35
            WHEN code_pattern = 'F_D3' THEN 35
            WHEN code_pattern = 'F_L1' THEN 25
            WHEN code_pattern = 'F_L2' THEN 10
            WHEN code_pattern = 'F_L3' THEN 10
            ELSE 10
        END
    ) as geom
FROM classification;

-- Étape 4 : L'index spatial pour rendre les futurs calculs ultra-rapides
CREATE INDEX idx_zones_motifs_v3_geom ON zones_motifs_filaires_v3 USING GIST(geom);

```

2- La profondeur du motif urbain (avant arbitrage)

Nouvelle définition d'une bande de motif urbain proportionnelle au rang hiérarchique des filaires. L'évolution de la profondeur fait varier l'intensité fonctionnelle.

[Le tableau de référence sous Drive](#)

Nouvelle version du 260305 - majoration des filaires sur les routes de desserte afin de mieux intersecter les items fonctionnels d'Openstreetmap. Dans cette version, la profondeur des filaires passe à 35 mètres pour les F_D1 et F_D2.

Hiérarchisation des patterns			Les Filaires		Géométrie OSM	Profondeur du motif urbain en mètres	Commentaires
				Contexte			
Filaire 1	Transit	F_T1	Corridor Métropolitain	Plus de 30 000 véhicules jours - 0 trottoirs	Higway	100	Vérifier
		F_T2	Voie Rapide Urbaine	Profil autoroutier à usage local	Ring Road	75	Abords autoroutiers (à vérifier)
		F_T3	Axe Structurant	Colonne vertébrale de type tram/bHNS + fonctionnalités	Multi-Way Boulevard	50	Exemples à développer
Filaire 2	Desserte	F_D1	Avenue Commercante	Axe à arbitrage modal nécessaire entre circulations automobile et autres	Downton streets	30	Tissu Haute Ville de Toulon
		F_D2	Rue marchande	Trafic automobile soutenu combiné avec réseau TC et vitalité commerciale forte	Shopping street	35	Contextualiser - St Jean / Pont du Las - François Cuzin/ XVeme Corps
		F_D3	La Piétonne	Mail piétonnier de grande ampleur (géométries) irrigué par TC et modes doux	Ramblas	35	Cours Lafayette

Filaire 3	Local	F_L1	Rue de quartier	Tissu urbain standard avec stationnement filaire et trottoirs accueillant en discontinu des fonctions urbaines		25	Loubière
		F_L2	Rue résidentielle	Voie routière sans TC ni fonctions urbaines regroupées		10	
		F_L3	Allées et sentes	cheminements strictement piétons		10	

DROP TABLE IF EXISTS zones_motifs_filaires_v2 CASCADE;

CREATE TABLE zones_motifs_filaires_v2 AS

WITH lignes_originales AS (

SELECT f.osm_id, f.nom_voie, f.code_pattern, ST_Transform(l.way, 3857) as geom_ligne

FROM zones_motifs_filaires f

JOIN planet_osm_line l ON f.osm_id = l.osm_id

)

SELECT

osm_id, nom_voie, code_pattern,

ST_Buffer(geom_ligne,

CASE

WHEN code_pattern = 'F_T1' THEN 100

WHEN code_pattern = 'F_T2' THEN 75

WHEN code_pattern = 'F_T3' THEN 50

WHEN code_pattern = 'F_D1' THEN 30

WHEN code_pattern = 'F_D2' THEN 35 -- Modifié

WHEN code_pattern = 'F_D3' THEN 35 -- Modifié

WHEN code_pattern = 'F_L1' THEN 25 -- Modifié

WHEN code_pattern = 'F_L2' THEN 10

WHEN code_pattern = 'F_L3' THEN 10

ELSE 10

END

) as geom

```
FROM lignes originales;

CREATE INDEX idx_zones_motifs_v2_geom ON zones_motifs_filaires_v2 USING GIST(geom);
```

Hiérarchisation des patterns			Les Filaires		Géométrie OSM	Profondeur du motif urbain en mètres	Commentaires
				Contexte			
Filaire 1	Transit	F_T1	Corridor Métropolitain	Plus de 30 000 véhicules jours - 0 trottoirs	Higway	100	Vérifier
		F_T2	Voie Rapide Urbaine	Profil autoroutier à usage local	Ring Road	75	Abords autoroutiers (à vérifier)

F_T3	Axe Structurant	Colonne vertébrale de type tram/bHNS + fonctionnalités	Multi-Way Boulevard	50	Exemples à développer		
Filaire 2	Desserte	F_D1	Avenue Commercante	Axe à arbitrage modal nécessaire entre circulations automobile et autres	Downton streets	30	Tissu Haute Ville de Toulon
		F_D2	Rue marchande	Trafic automobile soutenu combiné avec réseau TC et vitalité commerciale forte	Shopping street	25	Contextualiser - St Jean / Pont du Las - François Cuzin/ XVeme Corps
		F_D3	La Piétonne	Mail piétonnier de grande ampleur (géométries) irrigué par TC et modes doux	Ramblas	25	Cours Lafayette
Filaire 3	Local	F_L1	Rue de quartier	Tissu urbain standard avec stationnement filaire et trottoirs accueillant en discontinu des fonctions urbaines		15	Loubière
		F_L2	Rue résidentielle	Voie routière sans TC ni fonctions urbaines regroupées		10	

F_L3	Allées et sentes	cheminements strictement piétons		10	
------	------------------	----------------------------------	--	----	--

Le code SQL de Gemini

```
DROP TABLE IF EXISTS zones_motifs_filaires_v2;
```

```
CREATE TABLE zones_motifs_filaires_v2 AS
```

```
WITH lignes_originales AS (
```

```
-- On fait le lien entre vos codes Alteris et les lignes brutes d'OSM
```

```
SELECT
```

```
    f.osm_id,
```

```
    f.nom_voie,
```

```
    f.code_pattern,
```

```
    ST_Transform(l.way, 3857) as geom_ligne
```

```
FROM zones_motifs_filaires f
```

```
JOIN planet_osm_line l ON f.osm_id = l.osm_id
```

```
)
```

```
SELECT
```

```
    osm_id,
```

```
    nom_voie,
```

```
    code_pattern,
```

```
-- On applique les nouveaux buffers stricts de votre tableau Google Drive
```

```
ST_Buffer(geom_ligne,
```

```
    CASE
```

```
        WHEN code_pattern = 'F_T1' THEN 100
```

```
        WHEN code_pattern = 'F_T2' THEN 75
```

```
        WHEN code_pattern = 'F_T3' THEN 50
```

```
        WHEN code_pattern = 'F_D1' THEN 30
```

```
        WHEN code_pattern = 'F_D2' THEN 25
```

```
        WHEN code_pattern = 'F_D3' THEN 25
```

```
        WHEN code_pattern = 'F_L1' THEN 15
```

```
        WHEN code_pattern = 'F_L2' THEN 10
```

```
        WHEN code_pattern = 'F_L3' THEN 10
```

```
        ELSE 10 -- Sécurité par défaut
```

```
    END
```

```
) as geom
```

```
FROM lignes_originales;
```

```
CREATE INDEX idx_zones_motifs_v2_geom ON zones_motifs_filaires_v2 USING GIST(geom);
```

Le filaires sont conçues à partir d'un croisement de 3 paramètres urbains que sont

- la géométrie des voies sur OSM
- leur intensité capacitaire en matière d'écoulement des flux de mobilité
- le degré de densité fonctionnel

	Géométrie	Modalités cartographiques	Vitesse légale
N1	Autoroute / Voie Rapide	Absence de trottoir	Supérieure ou égal à 50 km/h
N2	Boulevard Métropolitain	Minimum de deux voies ouvertes à la circulation automobile	Comprise entre 50 et 70 km/h
N3	Avenue / Cours	Voie ouverte à la circulation automobile, avec stationnement filaire, TCSP possible, Largeur d'au moins 4,5 mètres minimum avec trottoir ou zone de rencontre, Présences de passages piétons,	maximum 50 km/heure
N4	Rue Faubourg / Principale	Voie ouverte à la circulation automobile inférieure à 4,5 mètres bordée par des fonctions urbaines, Sens unique possible,	maximum 30km/heure
N5	Rue de Desserte	Voie ouverte à la circulation automobile inférieure à 4,5 mètres bordée par des fonctions urbaines, Sens unique possible,	
N6	Ruelle / Impasse	Autres (bolléen)	
N7	Zones Piétonnes	Indicateur OSM	
	Capacitaire		

Niveau 1	Corridor de mobilité	Comptages routiers supérieurs à ? 000 véhicules/jour et irrigation par plusieurs lignes de transport en commun	
Niveau 2	Desserte majeure	Présence d'au moins une ligne TC	
Niveau 3	Desserte locale	Stationnement public filaire ou voie de mobilité douce	
	Fonctionnel		
Niveau 1	Centre Urbain	Présence continue ou presque d'activités commerciales / tertiaires / équipements publics / espaces publics	
Niveau 2	Centre Local	Agrégations de fonctionnalités commerciales ou tertiaires couplées à un intensité démographique INSEE Carroyé	
Niveau 3	Noyau de vie quotidienne	Ponctualité de services formant une centralité	
Niveau 4	Espaces Résidentiels	Restent des zones agglomérées OSM ne figurant dans aucun des 3 premiers niveaux	

Alteris propose une classification en 9 topologies de filaires

Hiérarchisation des patterns			Les Filaires		Géométrie OSM
				Contexte	
Filaire 1	Transit	F_T1	Corridor Métropolitain	Plus de 30 000 véhicules jours - 0 trottoirs	Higway
		F_T2	Voie Rapide Urbaine	Profil autoroutier à usage local	Ring Road
		F_T3	Axe Structurant	Colonne vertébrale de type tram/bHNS + fonctionnalités	Multi-Way Boulevard
Filaire 2	Desserte	F_D1	Avenue Commerçante	Axe à arbitrage modal nécessaire entre circulations automobile et autres	Downton streets
		F_D2	Rue marchande	Trafic automobile soutenu combiné avec réseau TC et vitalité commerciale forte	Shopping street
		F_D3	La Piétonne	Mail piétonnier de grande ampleur (géométries) irrigué par TC et modes doux	Ramblas
Filaire 3	Local	F_L1	Rue de quartier	Tissu urbain standard avec stationnement filaire et trottoirs accueillant en discontinu des fonctions urbaines	
		F_L2	Rue résidentielle	Voie routière sans TC ni fonctions urbaines regroupées	
		F_L3	Allées et sentes	cheminements strictement piétons	

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*([^\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 !