



RÉPUBLIQUE  
FRANÇAISE

*Liberté  
Égalité  
Fraternité*



# Documentation des Données foncières

Extrait de la documentation en ligne



DOCUMENTATION



MINISTÈRE  
DE LA TRANSITION  
ÉCOLOGIQUE  
ET DE LA COHÉSION  
DES TERRITOIRES

*Liberté  
Égalité  
Fraternité*



RÉPUBLIQUE  
FRANÇAISE

*Liberté  
Égalité  
Fraternité*



Le Cerema est un établissement public sous la tutelle du ministère de la Transition écologique, présent partout en métropole et dans les Outre-mer grâce à ses 26 implantations et ses 2 400 agents. Détenteur d'une expertise nationale mutualisée, le Cerema accompagne l'État et les collectivités territoriales pour la transition écologique, l'adaptation au changement climatique et la cohésion des territoires par l'élaboration coopérative, le déploiement et l'évaluation de politiques publiques d'aménagement et de transport. Doté d'un fort potentiel d'innovation et de recherche incarné notamment par son institut Carnot Clim'adapt, le Cerema agit dans 6 domaines d'activités : Expertise & ingénierie territoriale, Bâtiment, Mobilités, Infrastructures de transport, Environnement & Risques, Mer & Littoral.

Site web : [www.cerema.fr](http://www.cerema.fr)

## Avertissement

Ce document est extrait de la documentation en ligne, disponible sur le site <https://doc-datafoncier.cerema.fr/>, qui est mise à jour en continu. Ce document n'est donc valable qu'à un instant t, et le lecteur est invité à aller consulter la documentation en ligne pour bénéficier des dernières nouveautés.

Ce guide est un guide technique renseignant sur le contenu, les limites et la manière d'utiliser la base de données « Fichiers fonciers ». Pour d'autres questions (modalités d'accès, études réalisées...), le lecteur est invité à consulter le site <https://datafoncier.cerema.fr>

Cette documentation, évolutive, peut cependant encore contenir des erreurs ou coquilles. Si vous relevez une erreur dans celle-ci, n'hésitez pas à nous en faire part par mail via la boîte [datafoncier@cerema.fr](mailto:datafoncier@cerema.fr)



# Fiche

## Table des matières

### 1 Cas pratiques pour situer

#### 1.1 Cas pratique : Choisir sa période temporelle dans PostgreSQL/PostGIS

##### 1.1.1 Période annuelle

##### 1.1.2 Autre période temporelle

#### 1.2 Cas pratique : Situer les mutations dans PostgreSQL/PostGIS

##### 1.2.1 Trouver une mutation à partir d'une adresse

##### 1.2.2 Retrouver les adresses associées à une mutation

# 1 Cas pratiques pour situer

## 1.1 Cas pratique : Choisir sa période temporelle dans PostgreSQL/PostGIS

### 1.1.1 Période annuelle

Pour obtenir le nombre de mutations qui ont eu lieu sur le département du Nord (59) en 2013 :

```
SELECT count(*) as nombre_de_mutations_2013
FROM dvf.mutation
WHERE coddep = '59' AND anneemut = 2013;
```

Pour obtenir les mutations sur le département du Nord (59) sur la période 2010-2015 :

```
SELECT *
FROM dvf.mutation
WHERE coddep = '59' AND (anneemut >= 2010 AND anneemut <= 2015);
```

Pour obtenir le montant de transactions par année sur le département du Nord (59) :

```
SELECT anneemut, sum(valeurfonc) as montant_de_transactions
FROM dvf.mutation
WHERE coddep = '59'
GROUP BY anneemut
ORDER BY anneemut;
```

### 1.1.2 Autre période temporelle

Pour lister les mutations sur le département du Nord (59) entre le 9 avril 2014 et le 8 mai 2014 :

```
SELECT *
FROM dvf.mutation
WHERE coddep = '59' AND (datemut >= '2014-04-09' AND datemut <= '2014-05-08');
```

Pour obtenir le nombre et le montant de transactions par trimestre sur le département du Nord (59) :

```
SELECT anneemut || ' ' || trimestremut AS trimestre,
       count(*) as nombre_de_mutations,
       sum(valeurfonc) as montant_de_transactions
FROM
(
```



```

SELECT *,
CASE
WHEN moismut IN (1, 2, 3) THEN 'Trimestre 1'
WHEN moismut IN (4, 5, 6) THEN 'Trimestre 2'
WHEN moismut IN (7, 8, 9) THEN 'Trimestre 3'
WHEN moismut IN (10, 11, 12) THEN 'Trimestre 4'
END AS trimestremut
FROM dvf.mutation
WHERE coddep = '59'
)t
GROUP BY anneemut, trimestremut
ORDER BY anneemut, trimestremut;

```

## 1.2 Cas pratique : Situer les mutations dans PostgreSQL/PostGIS

### 1.2.1 Trouver une mutation à partir d'une adresse

Pour récupérer les mutations qui concernent le boulevard Victor Hugo à Lille (code insee : 59350) :

```

SELECT DISTINCT *
FROM dvf.mutation
WHERE coddep = '59' AND idmutation IN (
SELECT idmutation FROM dvf.local WHERE coddep = '59' AND ffcodinsee = '59350' AND ffvoie LIKE
'%BD%VICTOR%HUGO'
UNION
SELECT idmutation FROM dvf.disposition_parcelle WHERE coddep = '59' AND ffcodinsee = '59350' AND
ffvoie LIKE '%BD%VICTOR%HUGO'
);

```

### 1.2.2 Retrouver les adresses associées à une mutation

Pour récupérer les adresses correspondant à la mutation dont l'identifiant est 123456:

```

SELECT DISTINCT COALESCE(ffnovoie::VARCHAR, '') ||
COALESCE(ffbtq, '') || ' ' ||
COALESCE(ffvoie, '') || ' ' ||
COALESCE(ffcommune, '') AS adresse
FROM dvf.local WHERE idmutation = 123456

UNION

SELECT DISTINCT COALESCE(ffnovoie::VARCHAR, '') ||
COALESCE(ffbtq, '') || ' ' ||
COALESCE(ffvoie, '') || ' ' ||

```

```
COALESCE(ffcommune, '') AS adresse  
FROM dvf.disposition_parcelle WHERE idmutation = 123456
```

Pour récupérer les adresses présentes dans la source DVF correspondant à la mutation dont l'identifiant est 123456:

```
SELECT COALESCE(novoie::VARCHAR, '') ||  
COALESCE(btq, '') || ' ' ||  
COALESCE(typvoie, '') || ' ' ||  
COALESCE(voie, '') || ' ' ||  
COALESCE(codepostal, '') || ' ' ||  
COALESCE(commune, '') AS adresse  
FROM dvf.adresse a  
JOIN  
(  
SELECT idadresse AS idadresse FROM dvf.adresse_dispoparc WHERE idmutation = 123456  
UNION  
SELECT idadresse FROM dvf.adresse_local WHERE idmutation = 123456  
) t  
ON a.idadresse = t.idadresse;
```