



**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 typbien

##### 1.0.1 Description

##### 1.0.2 Fiabilité - Limite

##### 1.0.3 Construction

# 1 typbien

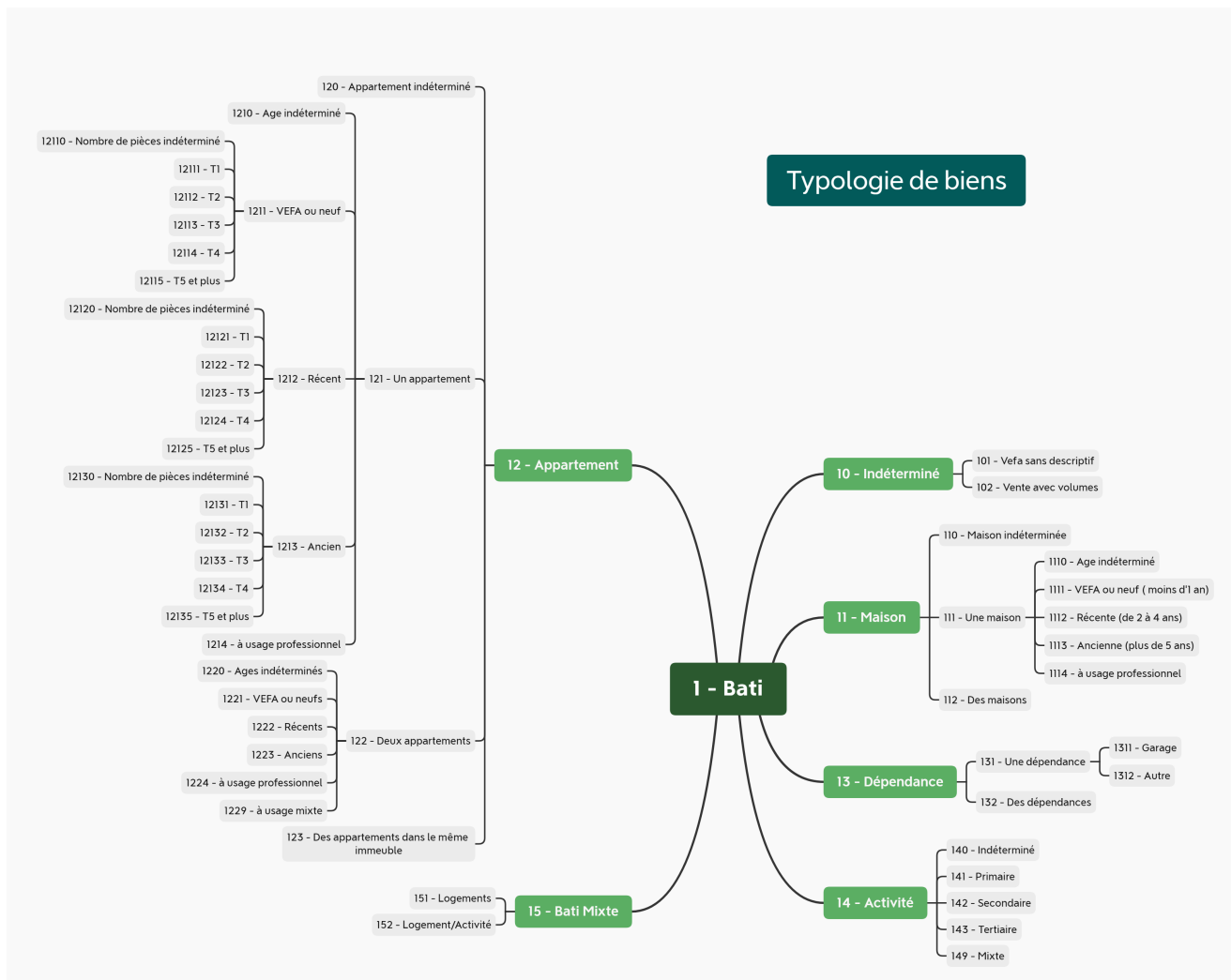
## 1.0.1 Description

Typologie de biens proposée par le Groupe national DVF (GnDVF). Elle permet de classer chacune des mutations selon un ensemble défini de catégories de bien.

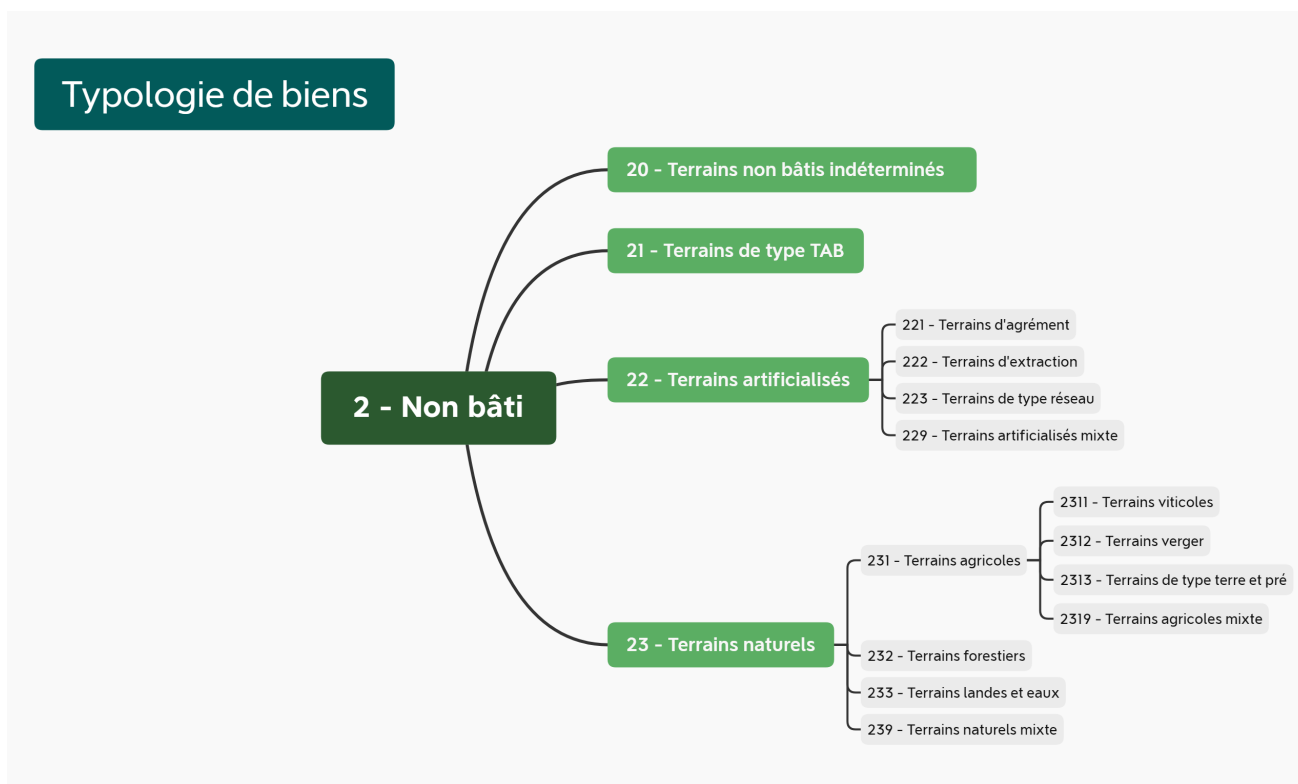
Cette typologie est différente de celle de la première version de DV3F.

Contrairement à la première typologie, elle s'appuie sur les données issues de DV3F (et non plus uniquement de DVF).

Les deux schémas suivants détaillent l'organisation en plusieurs niveaux de cette typologie :







## 1.0.2 Fiabilité - Limite

cf Caractériser une mutation à l'aide de la typologie de bien, Cette variable est systématiquement remplie.

Cette typologie peut être considérée comme fiable pour les biens immobiliers bâtis et en particulier les logements.

La typologie est à utiliser avec précaution pour tout ce qui concerne le foncier non bâti. Les catégories de biens proposés pour les mutations sans locaux sont à considérer comme une information à analyser plus finement, en particulier pour les terrains de « type TAB » qui ne correspondent qu'à une définition fiscale du terrain à bâtir.

Des typologies plus fines peuvent être rapidement construites en se basant sur celle-ci. Par exemple, pour les logements, il peut être intégré les périodes de construction, la présence de jardin, etc.

## 1.0.3 Construction

Cette variable est construite à partir de plusieurs champs de la table mutation.

### 1.0.3.1 Requête SQL

**WITH**

```

table_local AS (SELECT * FROM dvf.local),
-- tables necessaires pour la typologie de biens
mutation_avec_maison_ancienne AS (
SELECT DISTINCT idmutation
FROM table_local
WHERE codtyploc = 1 AND anciennete = 'ancien'
),
mutation_avec_maison_recente AS (
SELECT DISTINCT idmutation
FROM table_local
WHERE codtyploc = 1 AND anciennete = 'recent'
),
mutation_avec_maison_neuve AS (
SELECT DISTINCT idmutation
FROM table_local
WHERE codtyploc = 1 AND anciennete = 'neuf'
),
mutation_avec_appart_ancien AS (
SELECT DISTINCT idmutation
FROM table_local
WHERE codtyploc = 2 AND anciennete = 'ancien'
),
mutation_avec_appart_recent AS (
SELECT DISTINCT idmutation
FROM table_local
WHERE codtyploc = 2 AND anciennete = 'recent'
),
mutation_avec_appart_neuf AS (
SELECT DISTINCT idmutation
FROM table_local
WHERE codtyploc = 2 AND anciennete = 'neuf'
),
mutation_avec_2apparts_neufs AS (
SELECT idmutation
FROM table_local
GROUP BY idmutation
HAVING SUM(CASE WHEN codtyploc = 2 AND (anciennete = 'neuf') THEN 1 ELSE 0 END) = 2
),
mutation_avec_2apparts_recents AS (
SELECT idmutation
FROM table_local
GROUP BY idmutation
HAVING SUM(CASE WHEN codtyploc = 2 AND (anciennete = 'recent') THEN 1 ELSE 0 END) = 2
),
mutation_avec_2apparts_anciens AS (
SELECT idmutation
FROM table_local
GROUP BY idmutation
HAVING SUM(CASE WHEN codtyploc = 2 AND (anciennete = 'ancien') THEN 1 ELSE 0 END) = 2
),
mutation_dont_tous_les_apparts_meme_batiment AS ( -- mutation à partir de 3 appartements

```

```

SELECT idmutation
FROM (
  SELECT idmutation, array_agg(DISTINCT ffidbat) as I_ffidbat
  FROM table_local
  WHERE codtyploc = 2
  GROUP BY idmutation
  HAVING count(*) > 2
)t
WHERE array_length(I_ffidbat, 1) = 1 AND I_ffidbat[1] IS NOT NULL
),
mutation_avec_activite_secondaire AS (
  SELECT idmutation
  FROM table_local
  GROUP BY idmutation
  HAVING SUM(CASE WHEN codtyploc = 4 AND ffcnatloc IN ('U', 'US', 'UE', 'UG', 'UN') THEN 1 ELSE 0
END) > 0
  AND SUM(CASE WHEN codtyploc = 4 AND (ffcnatloc IS NULL OR ffcnatloc IN ('CA', 'CB', 'CM', 'CH',
'ME', 'CB', 'AT', 'AU', 'DC', 'SM', 'PP')) THEN 1 ELSE 0 END) = 0
),
mutation_avec_activite_tertiaire AS (
  SELECT idmutation
  FROM table_local
  GROUP BY idmutation
  HAVING SUM(CASE WHEN codtyploc = 4 AND ffcnatloc IN ('CA', 'CB', 'CM', 'CH', 'ME', 'CB', 'AT', 'AU',
'DC', 'SM', 'PP') THEN 1 ELSE 0 END) > 0
  AND SUM(CASE WHEN codtyploc = 4 AND (ffcnatloc IS NULL OR ffcnatloc IN ('U', 'US', 'UE', 'UG',
'UN')) THEN 1 ELSE 0 END) = 0
)
SELECT t1.*,
CASE
-- BATI (1)
WHEN nblocmut > 0 or vefa IS TRUE OR nbvolmut > 0 THEN
CASE
-- BATI - INDETERMINE (10) : ventes avec volume ou vefa sans information sur le local
WHEN (nblocmut = 0 AND vefa IS TRUE) OR nbvolmut > 0 THEN
CASE
-- BATI - INDETERMINE : vente avec volume(s) (102)
WHEN nbvolmut > 0 THEN '102'
-- BATI - INDETERMINE : Vefa sans descriptif (101)
WHEN (nblocmut = 0 AND vefa IS TRUE) THEN '101'
-- Ne doit pas exister
ELSE '100'
END
-- MAISON (11)
WHEN nblocmai > 0 and nblocapt = 0 AND nblocact = 0 THEN
CASE
-- MAISON - INDETERMINEE : la surface batie est inférieure à 9 m2 (110)
WHEN nblocmai = 1 AND sbati < 9 THEN '110'
-- MAISON INDIVIDUELLE (vendue seule) (111)
WHEN nblocmai = 1 THEN
CASE
-- MAISON à usage exclusivement professionnel (1114)

```



```

    WHEN ffnbloch = 0 THEN '1114'
  ---- Pour les catégories ci apres, il peut y avoir un peu d'usage professionnel (ffspro > 0)
  -- MAISON INDIVIDUELLE VEFA OU NEUF (1111)
    WHEN vefa IS TRUE OR t3.idmutation IS NOT NULL THEN '1111'
  -- MAISON INDIVIDUELLE RECENTE (1112)
    WHEN t2bis.idmutation IS NOT NULL THEN '1112'
  -- MAISON INDIVIDUELLE ANCIENNE (1113)
    WHEN t2.idmutation IS NOT NULL THEN '1113'
  -- MAISON INDIVIDUELLE INDETERMINEE (1110) (les informations des Fichiers fonciers ne
  sont pas disponibles ou non renseignés pour la date de construction)
    ELSE '1110'
  END
  -- MAISONS INDIVIDUELLES (112)
    WHEN nblocmai > 1 THEN '112'
    ELSE 'PROBLEME'
  END

-- APPARTEMENT (12)
  WHEN nblocapt > 0 and nblocmai = 0 AND nblocact = 0 THEN
  CASE
  -- 1 APPARTEMENT (121)
    WHEN nblocapt = 1 AND sbati >= 9 THEN

  CASE
  -- 1 APPARTEMENT à usage exclusivement professionnel (1214)
    WHEN ffnbloch = 0 THEN '1214'
  ---- Pour les catégories ci apres, il peut y avoir un peu d'usage professionnel (ffspro > 0)
  -- 1 APPARTEMENT VEFA OU NEUF (1211)
    WHEN vefa IS TRUE OR t5.idmutation IS NOT NULL THEN
    CASE
      -- 1 APPARTEMENT VEFA OU NEUF T1 (12111)
        WHEN nbapt1pp = 1 THEN '12111'
      -- 1 APPARTEMENT VEFA OU NEUF T2 (12112)
        WHEN nbapt2pp = 1 THEN '12112'
      -- 1 APPARTEMENT VEFA OU NEUF T3 (12113)
        WHEN nbapt3pp = 1 THEN '12113'
      -- 1 APPARTEMENT VEFA OU NEUF T4 (12114)
        WHEN nbapt4pp = 1 THEN '12114'
      -- 1 APPARTEMENT VEFA OU NEUF T5 ou + (12115)
        WHEN nbapt5pp = 1 THEN '12115'
      -- 1 APPARTEMENT VEFA OU NEUF nombre de pièce indéterminé (12110)
        ELSE '12110'
    END
  -- 1 APPARTEMENT RECENT (1212)
    WHEN t4bis.idmutation IS NOT NULL THEN
    CASE
      -- 1 APPARTEMENT RECENT T1 (12121)
        WHEN nbapt1pp = 1 THEN '12121'
      -- 1 APPARTEMENT RECENT T2 (12122)
        WHEN nbapt2pp = 1 THEN '12122'
      -- 1 APPARTEMENT RECENT T3 (12123)
        WHEN nbapt3pp = 1 THEN '12123'
    
```

```

-- 1 APPARTEMENT RECENT T4 (12124)
  WHEN nbapt4pp = 1 THEN '12124'
-- 1 APPARTEMENT RECENT T5 ou + (12125)
  WHEN nbapt5pp = 1 THEN '12125'
-- 1 APPARTEMENT RECENT nombre de pièce indéterminé (12120)
  ELSE '12120'
END
-- 1 APPARTEMENT ANCIEN (1213)
  WHEN t4.idmutation IS NOT NULL THEN
  CASE
-- 1 APPARTEMENT ANCIEN T1 (12131)
    WHEN nbapt1pp = 1 THEN '12131'
-- 1 APPARTEMENT ANCIEN T2 (12132)
    WHEN nbapt2pp = 1 THEN '12132'
-- 1 APPARTEMENT ANCIEN T3 (12133)
    WHEN nbapt3pp = 1 THEN '12133'
-- 1 APPARTEMENT ANCIEN T4 (12134)
    WHEN nbapt4pp = 1 THEN '12134'
-- 1 APPARTEMENT ANCIEN T5 ou + (12135)
    WHEN nbapt5pp = 1 THEN '12135'
-- 1 APPARTEMENT ANCIEN nombre de pièce indéterminé (12130)
    ELSE '12130'
  END
-- 1 APPARTEMENT INDETERMINE (1210) (les informations des Fichiers fonciers ne sont pas
disponibles ou non renseignées pour la date de construction)
  ELSE '1210'
END
-- 2 APPARTEMENTS (122)
  WHEN nblocapt = 2 THEN

  CASE
-- 2 APPARTEMENTS à usage exclusivement professionnel (1224)
    WHEN ffnbloch = 0 THEN '1224'
-- 2 APPARTEMENTS à usage mixte (1229) : un seul des appartements vendu a un usage
exclusivement professionnel
    WHEN ffnbloch = 1 THEN '1229'
---- Pour les catégories ci apres, il peut y avoir un peu d'usage professionnel (ffspro > 0)
-- 2 APPARTEMENTS en VEFA OU NEUFS (1221)
    WHEN vefa IS TRUE OR t7.idmutation IS NOT NULL THEN '1221'
-- 2 APPARTEMENTS RECENTS (1222)
    WHEN vefa IS TRUE OR t6bis.idmutation IS NOT NULL THEN '1222'
-- 2 APPARTEMENTS ANCIENS (1223)
    WHEN t6.idmutation IS NOT NULL THEN '1223'
-- 2 APPARTEMENTS INDETERMINES (1220) (les appartements sont habités avec des
anciennetés différentes /
-- les informations des Fichiers fonciers ne sont pas totalement disponibles ou
partiellement renseignées pour la date de construction)
    ELSE '1220'
  END
-- APPARTEMENTS dans le même bâtiment (123)
  WHEN nblocapt > 2 AND t8.idmutation IS NOT NULL THEN '123'

```

```

-- APPARTEMENTS INDETERMINES (1 appartement de - de 9 m2 / plusieurs apparts dans
plusieurs bâtiments / non rapatriement des Fichiers fonciers) (120)

```

```

  ELSE '120'

```

```

  END

```

```

-- DEPENDANCE (13)

```

```

  WHEN nblocdep > 0 AND nblocapt = 0 and nblocmai = 0 AND nblocact = 0 THEN

```

```

    CASE

```

```

      -- UNE DEPENDANCE (131)

```

```

        WHEN nblocdep = 1 THEN

```

```

          CASE

```

```

            -- 1 DEPENDANCE GARAGE (1311)

```

```

              WHEN ffnbpgarag > 0 THEN '1311'

```

```

            -- 1 DEPENDANCE AUTRE (1312)

```

```

              ELSE '1312'

```

```

          END

```

```

      -- DEPENDANCES (132)

```

```

        WHEN nblocdep > 1 THEN '132'

```

```

        ELSE 'PROBLEME'

```

```

    END

```

```

-- ACTIVITE (14)

```

```

  WHEN nblocact > 0 AND nblocapt = 0 and nblocmai = 0 THEN

```

```

    CASE

```

```

      -- ACTIVITE PRIMAIRE (141)

```

```

        -- IMPOSSIBLE A DETERMINER A CE STADE

```

```

      -- ACTIVITE SECONDAIRE (142)

```

```

        WHEN t9.idmutation IS NOT NULL THEN '142'

```

```

      -- ACTIVITE TERTIAIRE (143)

```

```

        WHEN t10.idmutation IS NOT NULL THEN '143'

```

```

      -- ACTIVITE MIXTE (149)

```

```

        WHEN nblocact = ffnblocact THEN '149'

```

```

      -- ACTIVITE INDETERMINE (140) (non rapatrié des Fichiers fonciers)

```

```

        ELSE '140'

```

```

    END

```

```

-- MIXTE BATI (15)

```

```

  -- MIXTE - LOGEMENTS (151)

```

```

  WHEN nblocact = 0 AND nblocmai > 0 AND nblocapt > 0 THEN '151'

```

```

  -- MIXTE - LOGEMENT/ACTIVITE (152)

```

```

  WHEN nblocact > 0 AND (nblocmai > 0 OR nblocapt > 0) THEN '152'

```

```

  ELSE 'PROBLEME'

```

```

  END

```

```

-- NON BATI (2)

```

```

ELSE

```

```

  CASE

```

```

    -- TERRAINS DE TYPE TAB (21)

```

```

      WHEN l_dcant[10] > 0

```

```

        OR libnatmut = 'Vente terrain à bâtir'

```

```

      OR

```

```

      l_artcgi && ARRAY['257-7-1*3','278 sexes l.1','691bis', '1594OGA']::VARCHAR[]

```

```

        -- depuis V4, on supprime '1594D*2

```

```

      THEN '21'

```

```

    -- TERRAINS ARTIFICIALISES (22)

```

```

      WHEN l_dcant[7] + l_dcant[9] + l_dcant[11] + l_dcant[12] + l_dcant[13] = sterr THEN

```

```

CASE
-- TERRAINS AGREEMENTS (221)
  WHEN I_dcnc[9] + I_dcnc[11] = sterr THEN '221'
-- TERRAINS D'EXTRACTION (222)
  WHEN I_dcnc[7] = sterr THEN '222'
-- TERRAINS DE TYPE RESEAU (223)
  WHEN I_dcnc[12] = sterr THEN '223'
-- TERRAINS ARTIFICIALISES MIXTES (229)
  ELSE '229'
END
-- TERRAINS NATURELS (23)
  WHEN I_dcnc[1] + I_dcnc[2] + I_dcnc[3] + I_dcnc[4] + I_dcnc[5] + I_dcnc[6] + I_dcnc[8] = sterr THEN
-- TERRAINS AGRICOLES (231)
  CASE
-- TERRAINS VITICOLES (2311)
    WHEN I_dcnc[4] >= sterr * 0.25 THEN '2311'
-- TERRAINS VERGERS (2312)
    WHEN I_dcnc[3] >= sterr * 0.35 THEN '2312'
-- TERRAINS DE TYPE TERRES ET PRES (2313)
    WHEN I_dcnc[1] + I_dcnc[2] >= sterr * 0.40 THEN '2313'
-- TERRAINS AGRICOLES MIXTES (2319)
    WHEN I_dcnc[1] + I_dcnc[2] + I_dcnc[3] + I_dcnc[4] >= sterr * 0.40 THEN '2319'
-- TERRAINS FORESTIERS (232)
    WHEN I_dcnc[5] > sterr * 0.60 THEN '232'
-- TERRAINS LANDES ET EAUX (233)
    WHEN I_dcnc[6] + I_dcnc[8] > sterr * 0.60 THEN '233'
-- TERRAINS NATURELS MIXTES (239)
    ELSE '239'
  END
-- TERRAINS NON BATIS INDETERMINEES (20)
  ELSE '20'
END
END as typologie_de_bien
FROM dvf.mutation t1
LEFT JOIN mutation_avec_maison_ancienne t2
ON t1.idmutation = t2.idmutation
LEFT JOIN mutation_avec_maison_recente t2bis
ON t1.idmutation = t2bis.idmutation
LEFT JOIN mutation_avec_maison_neuve t3
ON t1.idmutation = t3.idmutation
LEFT JOIN mutation_avec_appart_ancien t4
ON t1.idmutation = t4.idmutation
LEFT JOIN mutation_avec_appart_recent t4bis
ON t1.idmutation = t4bis.idmutation
LEFT JOIN mutation_avec_appart_neuf t5
ON t1.idmutation = t5.idmutation
LEFT JOIN mutation_avec_2apparts_anciens t6
ON t1.idmutation = t6.idmutation
LEFT JOIN mutation_avec_2apparts_recents t6bis
ON t1.idmutation = t6bis.idmutation
LEFT JOIN mutation_avec_2apparts_neufs t7
ON t1.idmutation = t7.idmutation

```

**LEFT JOIN mutation\_dont\_tous\_les\_apparts\_meme\_batiment t8**  
**ON t1.idmutation = t8.idmutation**  
**LEFT JOIN mutation\_avec\_activite\_secondaire t9**  
**ON t1.idmutation = t9.idmutation**  
**LEFT JOIN mutation\_avec\_activite\_tertiaire t10**  
**ON t1.idmutation = t10.idmutation**