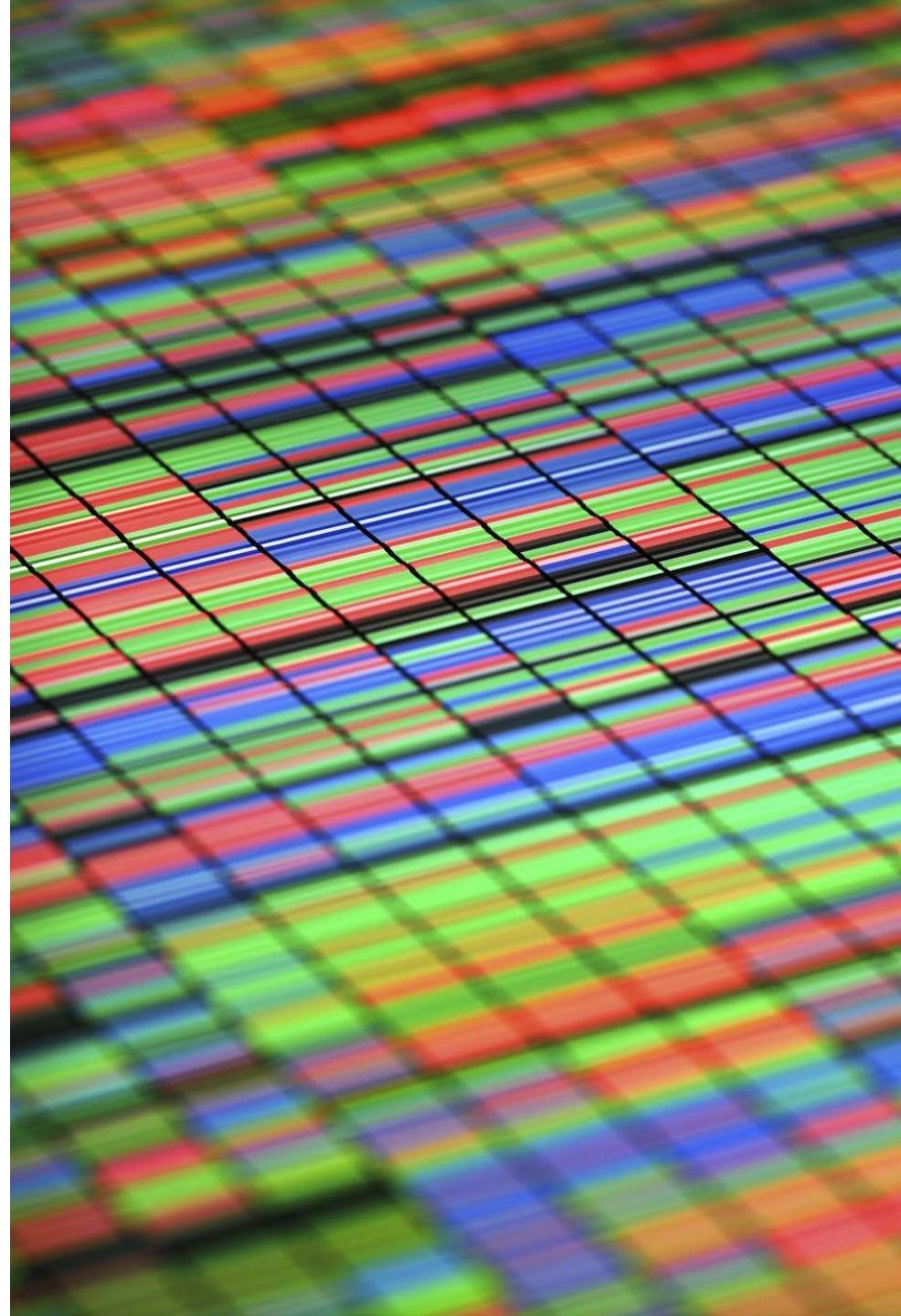


Conception d'une base de données immobilières

Naima Boutrah

Février 2026



Contexte du projet

Création d'une base de données permettant de collecter les transactions immobilières et foncières en France pour analyser le marché et aider les différentes agences régionales à mieux accompagner leurs clients.

Les données initiales & la conformité RGPD

Données brutes

- **donnees_communes.xlsx** :
contient les informations sur
les communes
- **fr-esr-referentiel-
géographique.xlsx** : contient
des informations sur les
régions et départements
- **Valeurs-foncières.xlsx** :
contient des informations sur
les transactions foncières

Conformité RGPD

J'ai exclu la colonne "**Nom de l'acquéreur**" du fichier Valeurs-foncières.xlsx car le stockage de **noms de particuliers** associés à leur patrimoine immobilier et **leur adresse** sans leur consentement est une violation grave du RGPD (Règlement Général sur la Protection des Données)

La norme 3NF

Un modèle de données basé sur la **3^{ème} Formes Normales (3NF)** et **d'intégrité référentielle** => une colonne ne doit dépendre que de la clé primaire de sa table

L'extrait du dictionnaire des données

Entité : Région (C'est la table de base pour la hiérarchie géographique)						
Champ (Code)	Description	Type SQL/ longeur	Règle de gestion	REGLE DE CALCUL	Source Fichier	
code_reg	Code unique de la région	VARCHAR(3)	NOT NULL (PK)		referentiel-geo (reg_code)	
nom_region	Nom complet de la région	VARCHAR(50)	NOT NULL		referentiel-geo (reg_nom)	
regrgp_nom	Nom regroupement régions	VARCHAR(50)	NOT NULL		referentiel-geo (regrgp_nom)	

Entité : Département (Lié à la région. Un département ne peut exister sans région)					
Champ (Code)	Description	Type SQL	Règle de gestion	REGLE DE CALCUL	Source Fichier
code_dep	Code du département (ex: 01, 2A)	VARCHAR(3)	NOT NULL (PK)		referentiel-geo (dep_code)
nom_dep	Nom du département	VARCHAR(50)	NOT NULL		referentiel-geo (dep_nom)
#code_reg	Référence à la table REGION	VARCHAR(3)	NOT NULL (FK)		referentiel-geo (code_reg)

Entité : Commune (regroupe les infos de population et les liens administratifs)					
Champ (Code)	Description	Type SQL	Règle de gestion	REGLE DE CALCUL	Source Fichier
code_dep_com	code commune	VARCHAR(5)	NOT NULL (PK)	concaténation code dep + code commune	donnees_communes (CODDEP + CODCOM)
nom_com	Nom de la commune	VARCHAR(100)	NOT NULL		donnees_communes (COM)
code_postal	code postal	VARCHAR(5)	NULL		
pop_totale	Population totale de la commune	INT	NOT NULL		donnees_communes (PTOT)
#code_dep	Référence au Département	VARCHAR(3)	NOT NULL(FK)		donnees_communes (CODDEP)

Entité : Bien (décrit l'objet immobilier de manière unique)

Champ (Code)	Description	Type SQL	Règle de gestion	REGLE DE CALCUL	Source Fichier
id_bien	Identifiant unique auto-incrémenté	SERIAL	NOT NULL (PK)		Création SQL
no_voie	Numéro dans la voie	INT	NULL		Valeurs-foncières (No voie)
BTQ	Indice (Bis/Ter/Quater)	VARCHAR(3)	NULL		Valeurs-foncières (B/T/Q)
type_voie	Type (Rue, Avenue, etc.)	VARCHAR(50)	NULL		Valeurs-foncières (Type de voie)
voie	Nom de la rue/voie	VARCHAR(250)	NOT NULL		Valeurs-foncières (Voie)
nb_pieces	Nombre de pièces principales	INT	NOT NULL		Valeurs-foncières (Nombre pieces principales)
Surface_Carrez	Surface Carrez du bâti	FLOAT	NOT NULL	m²	Valeurs-foncières (Surface Carrez)
surface_bati	Surface réelle du bâti	INT	NOT NULL	m²	Valeurs-foncières (Surface reelle bati)
type_local	Appartement, Maison, Dépendance	VARCHAR(50)	NOT NULL		Valeurs-foncières (Type local)
code_dep_com	Référence à Commune	VARCHAR(5)	NOT NULL (FK)		Valeurs-foncières (concat_dep_com)

Entité : Vente(historique des transactions liées aux biens)

Champ (Code)	Description	Type SQL	Règle de gestion	REGLE DE CALCUL	Source Fichier
id_vente	Identifiant unique de transaction	SERIAL	NOT NULL (PK)		Création SQL
date_mutation	Date de la vente	DATE	NOT NULL		Valeurs-foncières (Date mutation)
valeur_fonciere	Montant de la transaction	BIGINT	NOT NULL	€	Valeurs-foncières (Valeur fonciere)
nature_vente	Type (Vente, Échange, Adjudication)	VARCHAR(50)	NOT NULL		Valeurs-foncières (Nature mutation)
#id_bien	Référence au BIEN concerné	INT	NOT NULL (FK)		Lien interne
Nom de l'acquéreur	Champ supprimé au chargement (Anonymisation RGPD)				

Le schéma relationnel normalisé

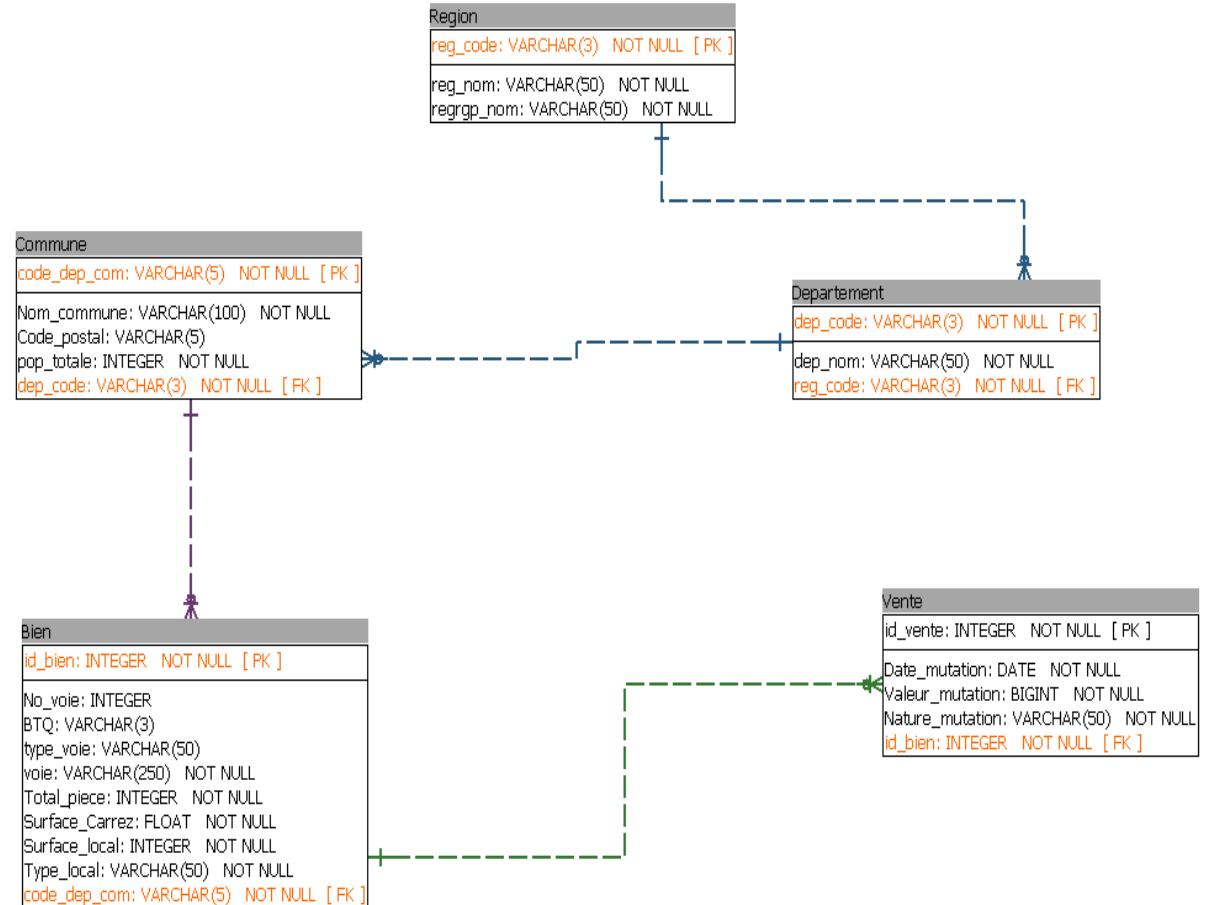
❖ Les Cardinalités

region (1) ---- (N) departement : Une région a plusieurs départements, un département n'est que dans une région.

departement (1) ---- (N) commune : Un département contient plusieurs communes. une commune appartient à un seul département

Commune (1) ---- (N) bien : Une commune possède plusieurs biens immobiliers. Un bien appartient à une seule commune

bien (1) ---- (N) vente : Un bien peut avoir été vendu plusieurs fois au fil des années (historique des prix). Mais une vente spécifique ne concerne qu'un seul bien.



La base de données avec les tables créées et les données chargées

The image shows two screenshots of the pgAdmin 4 interface. The left screenshot displays the 'Object Explorer' tree. It shows a 'Servers' node, a 'PostgreSQL 18' node under it, and a 'Databases(4)' node. A database named 'data_immo' is highlighted with an oval. The right screenshot shows the 'Statistics' tab of a table information page. The table has six columns: 'Table name', 'Total Size', 'Tuples inserted', 'Tuples updated', 'Tuples deleted', and 'Tuples HOT updated'. The data for five tables is listed:

Table name	Total Size	Tuples inserted	Tuples updated	Tuples deleted	Tuples HOT updated
bien	4 MB	34169	0	0	0
commune	2.94 MB	34991	0	0	0
departement	24 KB	109	0	0	0
region	24 KB	19	0	0	0
vente	2.75 MB	34169	0	0	0

- **Choix SGBDR PostgreSQL** : Open Source, gestion robuste du SQL, gestion gros volume de données

Code SQL

```
CREATE TABLE public.Region (
    reg_nom VARCHAR(50) NOT NULL,
    regrgp_nom VARCHAR(50) NOT NULL,
    CONSTRAINT id_region PRIMARY KEY (reg_code)
);

CREATE TABLE public.Departement (
    dep_code VARCHAR(3) NOT NULL,
    dep_nom VARCHAR(50) NOT NULL,
    reg_code VARCHAR(3) NOT NULL,
    CONSTRAINT departement_pk PRIMARY KEY (dep_code)
);

CREATE TABLE public.Commune (
    code_dep_com VARCHAR(5) NOT NULL,
    Nom_commune VARCHAR(100) NOT NULL,
    Code_postal VARCHAR(5),
    pop_totale INTEGER NOT NULL,
    dep_code VARCHAR(3) NOT NULL,
    CONSTRAINT id_codedep_codecommune PRIMARY KEY (code_dep_com)
);

CREATE SEQUENCE public.bien_id_bien_seq_1;
```

```
CREATE TABLE public.Bien (
    id_bien INTEGER NOT NULL DEFAULT nextval('public.bien_id_bien_seq_1')
        No_voie INTEGER,
        BTQ VARCHAR(3),
        type_voie VARCHAR(50),
        voie VARCHAR(250) NOT NULL,
        Total_piece INTEGER NOT NULL,
        Surface_Carrez NUMERIC(10,2) NOT NULL,
        Surface_local NUMERIC(10,2) NOT NULL,
        Type_local VARCHAR(50) NOT NULL,
        code_dep_com VARCHAR(5) NOT NULL,
    CONSTRAINT id_bien PRIMARY KEY (id_bien)
);

ALTER SEQUENCE public.bien_id_bien_seq_1 OWNED BY public.Bien.id_bien;
CREATE SEQUENCE public.vente_id_vente_seq;
CREATE TABLE public.Vente (
    id_vente INTEGER NOT NULL DEFAULT nextval('public.vente_id_vente_seq'
        Date_mutation DATE NOT NULL,
        Valeur_mutation NUMERIC(15,2) NOT NULL,
        Nature_mutation VARCHAR(50) NOT NULL,
        id_bien INTEGER NOT NULL,
    CONSTRAINT id_vente PRIMARY KEY (id_vente)
);
```

Code SQL (suite)

```
ALTER SEQUENCE public.vente_id_vente_seq OWNED BY public.Vente.id_vente;
ALTER TABLE public.Departement ADD CONSTRAINT region_departement_fk
FOREIGN KEY (reg_code)
REFERENCES public.Region (reg_code)
ON DELETE NO ACTION
ON UPDATE NO ACTION
NOT DEFERRABLE;

ALTER TABLE public.Commune ADD CONSTRAINT departement_communne_fk
FOREIGN KEY (dep_code)
REFERENCES public.Departement (dep_code)
ON DELETE NO ACTION
ON UPDATE NO ACTION
NOT DEFERRABLE;
```

```
ALTER TABLE public.Bien ADD CONSTRAINT commune_bien_fk
FOREIGN KEY (code_dep_com)
REFERENCES public.Commune (code_dep_com)
ON DELETE NO ACTION
ON UPDATE NO ACTION
NOT DEFERRABLE;

ALTER TABLE public.Vente ADD CONSTRAINT bien_vente_fk
FOREIGN KEY (id_bien)
REFERENCES public.Bien (id_bien)
ON DELETE NO ACTION
ON UPDATE NO ACTION
NOT DEFERRABLE;
```

Les requêtes permettant de démontrer le bon chargement des données

The screenshot shows a MySQL Workbench interface. The 'Query' tab is active, displaying the following SQL code:

```
1  SELECT
2      (SELECT COUNT(*) FROM region) AS nb_regions,
3      (SELECT COUNT(*) FROM departement) AS nb_departements,
4      (SELECT COUNT(*) FROM commune) AS nb_communes,
5      (SELECT COUNT(*) FROM bien) AS nb_biens,
6      (SELECT COUNT(*) FROM vente) AS nb_ventes;
```

Below the code, the 'Data Output' tab is selected, showing the results of the query:

	nb_regions	nb_departements	nb_communes	nb_biens	nb_ventes
1	19	109	34991	34169	34169

Table SQL	Nbr lignes	Fichiers.csv	Nbr lignes	Ecart
region	19	ref.géo	19	0
departement	109	ref.géo et communes	109	0
commune	34991	communes	34991	0
bien	34169	valeurs foncières	34169	0
vente	34169	valeurs foncières	34169	0

Requêtes SQL et résultats

R1 - Nombre total d'appartements vendus au 1er semestre 2020

-- 1)Nombre total d'appartements vendus au 1er semestre 2020

```
SELECT count(*) AS "Nb appart vendus"  
FROM vente  
  
JOIN bien ON bien.id_bien = vente.id_bien  
  
WHERE bien.type_local = 'Appartement'  
AND date_mutation BETWEEN '2020-01-01' AND '2020-06-30';
```



Nb appart vendus S1-2020
31378

R2- Le nombre de ventes d'appartement par région pour le 1er semestre 2020

-- 2)Le nombre de ventes d'appartement du 1er semestre 2020 par région

SELECT

```
region.reg_nom AS "Région",
count(*) AS "Nombre de ventes"
```

FROM vente

JOIN bien **ON** bien.id_bien = vente.id_bien

JOIN commune **ON** commune.code_dep_com = bien.code_dep_com

JOIN departement **ON** departement.dep_code = commune.dep_code

JOIN region **ON** region.reg_code = departement.reg_code

WHERE bien.type_local = 'Appartement'

AND date_mutation **BETWEEN** '2020-01-01' **AND** '2020-06-30'

GROUP BY region.reg_nom

ORDER BY "Nombre de ventes"

DESC ; -- Tri décroissant

Région	Nombre de ventes
Ile-de-France	13995
Provence-Alpes-Côte d'Azur	3649
Auvergne-Rhône-Alpes	3253
Nouvelle-Aquitaine	1932
Occitanie	1640
Pays de la Loire	1357
Hauts-de-France	1254
Grand Est	984
Bretagne	983
Normandie	862
Centre-Val de Loire	696
Bourgogne-Franche-Comté	376
Corse	223
Martinique	94
La Réunion	44
Guyane	34
Guadeloupe	2
Total	31378

R3- Proportion des ventes d'appartements par le nombre de pièces

-- 3)Proportion des ventes d'appartements par le nombre de pièces

SELECT

bien.total_piece **AS** Nb_pièce,

-- Calcul proportion/volume des ventes

COUNT(*) AS "Nbr_ventes",

ROUND(COUNT(*) *100 / SUM(COUNT(*)) OVER(), 2) AS "Prop_volume (%)",

-- Calcul proportion/montant des ventes

SUM(vente.valeur_mutation) AS "Mt_ventes",

ROUND(sum(vente.valeur_mutation) *100 / SUM(sum(vente.valeur_mutation))

OVER(), 2) AS "Prop_montant (%))"

FROM vente

JOIN bien **ON** bien.id_bien = vente.id_bien

WHERE bien.type_local = 'Appartement'

GROUP BY bien.total_piece

ORDER BY bien.total_piece

Nbr_pièces	Nbr_vente	Prop_volume (%)	Montant vente	Prop_montant (%)
0	30	0,10	3 337 249	0,04
1	6 739	21,48	969 036 638	12,34
2	9 783	31,18	1 983 921 879	25,27
3	8 966	28,57	2 425 437 082	30,89
4	4 460	14,21	1 507 375 435	19,2
5	1 114	3,55	587 777 021	7,49
6	204	0,65	204 260 160	2,6
7	54	0,17	99 419 213	1,27
8	17	0,05	55 731 860	0,71
9	8	0,03	14 065 142	0,18
10	2	0,01	657 000	0,01
11	1	0,00	139 000	0
TOTAUX	31 378	100,00	7 851 157 679	100

R4- Liste des 10 départements où le prix du mètre carré est le plus élevé

■ 4) Liste des 10 départements où le prix du mètre carré est le plus élevé.

SELECT

```
departement.dep_code AS "code département",
departement.dep_nom AS "Nom département",
sum(bien.surface_carrez) AS "Surface",
sum(vente.valeur_mutation) AS "Valeurs foncières",
```

ROUND((sum(vente.valeur_mutation) / sum(bien.surface_carrez)) , 2) **AS "Prix m²"**

FROM vente

JOIN bien **ON** bien.id_bien = vente.id_bien

JOIN commune **ON** commune.code_dep_com = bien.code_dep_com

JOIN departement **ON** departement.dep_code = commune.dep_code

WHERE bien.surface_carrez > 0 -- Évite la division par zéro

GROUP BY departement.dep_code, departement.dep_nom

ORDER BY "Prix m²"

LIMIT 10;

Code département	Nom département	Surface	Valeurs foncières	Prix m ²
75	Paris	249 933,12	2 973 890 342,00	11 898,74
92	Hauts-de-Seine	102 789,08	744 006 157,00	7 238,18
94	Val-de-Marne	110 975,56	536 164 441,00	4 831,37
6	Alpes-Maritimes	72 207,27	331 274 231,00	4 587,82
74	Haute-Savoie	19 627,58	81 635 375,00	4 159,22
93	Seine-Saint-Denis	56 755,04	230 731 268,00	4 065,39
78	Yvelines	101 990,73	408 182 949,00	4 002,16
69	Rhône	91 622,13	356 730 380,00	3 893,50
2A	Corse-du-Sud	14 569,15	54 674 522,00	3 752,76
33	Gironde	71 139,78	253 821 604,00	3 567,93

R5- Prix moyen du mètre carré d'une maison en Île-de-France

--5) Prix moyen du mètre carré d'une maison en Île-de-France

SELECT

region.reg_nom **AS** "Région",
ROUND(AVG (vente.valeur_mutation / bien.surface_carrez) , 2) **AS** "Prix moyen m²"

FROM vente

JOIN bien **ON** bien.id_bien = vente.id_bien
JOIN commune **ON** commune.code_dep_com = bien.code_dep_com
JOIN departement **ON** departement.dep_code = commune.dep_code
JOIN region **ON** region.reg_code = departement.reg_code

WHERE region.reg_nom = 'Île-de-France' **AND** bien.type_local = 'Maison'
GROUP BY region.reg_code ,region.reg_nom ;

Région	Prix moyen m ² (Maison)
Île-de-France	3997,71

R6- Liste des 10 appartements les plus chers avec la région et le nombre de mètres carrés.

--6)Liste des 10 appartements les plus chers avec la région et le nombre de mètres carrés.

SELECT

```
region.reg_nom AS "Région",
departement.dep_code AS "Code département",
departement.dep_nom AS « Nom département",
bien.surface_carrez AS "Surface carrez m2",
vente.valeur_mutation AS "Mt vente €",
ROUND(vente.valeur_mutation / bien.surface_local, 0) AS "Prix m2 (€)"
```

FROM vente

JOIN bien **ON** bien.id_bien = vente.id_bien

JOIN commune **ON** commune.code_dep_com = bien.code_dep_com

JOIN departement **ON** departement.dep_code = commune.dep_code

JOIN region **ON** region.reg_code = departement.reg_code

WHERE bien.type_local = 'Appartement'

ORDER BY vente.valeur_mutation **DESC**

LIMIT 10;

Région	Code département	Nom département	Surface carrez m ²	Mt vente €	Prix m ² (€)
Ile-de-France	75	Paris	9,10	9 000 000	989 011
Ile-de-France	91	Essonne	64,00	8 600 000	134 375
Ile-de-France	75	Paris	20,55	8 577 713	417 407
Ile-de-France	75	Paris	42,77	7 620 000	178 162
Ile-de-France	75	Paris	253,30	7 600 000	30 004
Ile-de-France	75	Paris	139,90	7 535 000	53 860
Ile-de-France	75	Paris	360,95	7 420 000	20 557
Ile-de-France	75	Paris	595,00	7 200 000	12 101
Ile-de-France	75	Paris	122,56	7 050 000	57 523
Ile-de-France	75	Paris	79,38	6 600 000	83 144

⚠️ **Attention :** Les données publiques peuvent contenir des erreurs de saisie ou des mutations complexes. La vigilance de l'analyste est la clé de la fiabilité.

R6- Liste des 10 appartements les plus chers (Script ajusté)

--6) Liste des 10 appartements les plus chers avec la région et le nombre de mètres carrés.

SELECT

```
region.reg_nom AS "Région",
departement.dep_code AS "Code département",
departement.dep_nom AS "Département",
bien.surface_carrez AS "Surface carrez m2",
vente.valeur_mutation AS "Mt vente",
ROUND(vente.valeur_mutation / bien.surface_carrez, 0) AS "Prix m2 (€)"
FROM vente
JOIN bien ON bien.id_bien = vente.id_bien
JOIN commune ON commune.code_dep_com = bien.code_dep_com
JOIN departement ON departement.dep_code = commune.dep_code
JOIN region ON region.reg_code = departement.reg_code
WHERE bien.type_local = 'Appartement'
-- On exclut les montants aberrants
AND (vente.valeur_mutation / bien.surface_carrez) < 40000
ORDER BY vente.valeur_mutation DESC
LIMIT 10;
```

Région	Code département	Nom département	Surface carrez m ²	Mt vente €	Prix m ² (€)
Ile-de-France	75	Paris	253,30	7 600 000	30 004
Ile-de-France	75	Paris	360,95	7 420 000	20 557
Ile-de-France	75	Paris	595,00	7 200 000	12 101
Ile-de-France	75	Paris	317,05	6 500 000	20 502
Ile-de-France	75	Paris	205,08	6 000 000	29 257
Ile-de-France	75	Paris	170,82	5 600 000	32 783
Ile-de-France	75	Paris	328,35	5 550 000	16 903
Ile-de-France	75	Paris	157,00	5 500 000	35 032
Ile-de-France	75	Paris	211,07	5 054 500	23 947
Ile-de-France	75	Paris	390,10	4 657 000	11 938

R7- Taux d'évolution du nombre de ventes entre le premier et le second trimestre de 2020

-- 7)Taux d'évolution du nombre de ventes entre le premier et le second trimestre de 2020.

WITH

```
    venteT1 AS (
        SELECT COUNT(*) AS NB_vente_T1
        FROM vente
        WHERE vente.date_mutation BETWEEN '2020-01-01' AND '2020-03-31'
    ),
```

```
    venteT2 AS (
        SELECT COUNT(*) AS NB_vente_T2
        FROM vente
        WHERE vente.date_mutation BETWEEN '2020-04-01' AND '2020-06-30'
    )
```

SELECT

```
    venteT1.NB_vente_T1 AS "Ventes T1",
    venteT2.NB_vente_T2 AS "Ventes T2",
    ROUND((venteT2.NB_vente_T2 - venteT1.NB_vente_T1) * 100.0 / venteT1.NB_vente_T1, 2)
        AS "Evolution(%)"
```

FROM venteT1,venteT2;

Ventes T1	Ventes T2	Evolution(%)
16 776	17 393	3,68

R8- Le classement des régions par rapport au prix au mètre carré des appartements de plus de 4 pièces

-- 8)Le classement des régions par rapport au prix au mètre carré des appartements de plus de 4 pièces.

SELECT

```
reg_nom AS " Nom région",
COUNT(vente.id_vente) AS "Nbr ventes",
round((sum(vente.valeur_mutation) / sum(bien.surface_carrez)),2) AS "prix
m2(F5 et +)"
```

FROM region

```
JOIN departement ON departement.reg_code = region.reg_code
JOIN commune ON commune.dep_code = departement.dep_code
JOIN bien ON bien.code_dep_com = commune.code_dep_com
JOIN vente ON vente.id_bien = bien.id_bien
```

```
WHERE bien.type_local = 'Appartement' AND bien.total_piece > 4
Group BY region.reg_nom
ORDER BY "Nbr ventes" DESC;
```

Nom région	Nbr ventes	prix m ² (F5 et +)
Ile-de-France	685	9171,26
Auvergne-Rhône-Alpes	185	2932,00
Provence-Alpes-Côte d'Azur	117	3166,99
Grand Est	76	1475,14
Nouvelle-Aquitaine	68	2545,21
Pays de la Loire	54	2495,17
Occitanie	47	2182,36
Hauts-de-France	40	2280,41
Bretagne	37	2325,81
Centre-Val de Loire	31	1459,09
Normandie	26	2119,37
Bourgogne-Franche-Comté	26	1125,27
Corse	5	3444,73
La Réunion	2	3469,55
Martinique	1	573,48

R9- Liste des communes ayant eu au moins 50 ventes au 1er trimestre

-- 9) Liste des communes ayant eu au moins 50 ventes au 1er trimestre

```
SELECT
    commune.nom_commune AS "Communes",
    COUNT(*) AS "NB ventes"
FROM vente
JOIN bien ON bien.id_bien = vente.id_bien
JOIN commune ON commune.code_dep_com = bien.code_dep_com
WHERE vente.date_mutation BETWEEN '2020/01/01' AND '2020/03/31'

-- On regroupe par id_com pour ne pas regrouper les communes qui ont le même nom
GROUP BY commune.code_dep_com, commune.nom_commune
HAVING COUNT(*) >= 50
ORDER BY "NB ventes" DESC;
```

Data Output Messages Notifications

The screenshot shows a database interface with a toolbar at the top containing icons for new table, open table, save, refresh, delete, export, and search. Below the toolbar is a table with two columns: 'Communes' (character varying (100)) and 'NB ventes' (bigint). The table lists seven entries, all from Paris, with counts ranging from 157 to 228. At the bottom of the table, it says 'Total rows: 48' and 'Query complete 00:00:00.171'.

	Communes character varying (100)	NB ventes bigint
1	Paris 17e Arrondissement	228
2	Paris 15e Arrondissement	215
3	Paris 18e Arrondissement	209
4	Nice	173
5	Paris 11e Arrondissement	169
6	Paris 16e Arrondissement	165
7	Bordeaux	157

Total rows: 48 Query complete 00:00:00.171

R10- Différence en pourcentage du prix au mètre carré entre un appartement de type F2 et F3

-- 10)Différence en pourcentage du prix au mètre carré entre un appartement de 2 pièces et un appartement de 3 pièces.

SELECT

```
    prix_t2 AS "Prix m2/T2",
    prix_t3 AS "Prix m2/T3",
    ROUND((prix_t2 - prix_t3) * 100.0 / prix_t3, 2) AS "Ecart %"
```

FROM (

SELECT

```
    ROUND(SUM(CASE WHEN total_piece = 2 THEN valeur_mutation END) / SUM(CASE WHEN total_piece = 2 THEN surface_carrez END), 2) AS prix_t2,
    ROUND(SUM(CASE WHEN total_piece = 3 THEN valeur_mutation END) / SUM(CASE WHEN total_piece = 3 THEN surface_carrez END), 2) AS prix_t3
FROM vente
JOIN bien ON bien.id_bien = vente.id_bien
WHERE bien.type_local = 'Appartement' AND bien.total_piece IN (2,3)
```

) **AS** calcul_global;

Prix m ² /T2	Prix m ² /T3	Ecart %
4727,78	4135,75	14,31

R11- Les moyennes de valeurs foncières pour le top 3 des communes des départements 6, 13, 33, 59 et 69

```
-- 11)Les moyennes de valeurs foncières pour le top 3 des communes  
-- des départements 6, 13, 33, 59 et 69
```

```
WITH  
valeur_par_ville AS (  
SELECT dep_code, nom_commune, avg(valeur_mutation) as "valeur"  
FROM vente  
JOIN bien USING (id_bien)  
JOIN commune USING (code_dep_com)  
WHERE commune.dep_code IN ('06', '13', '33', '59', '69')  
GROUP BY commune.dep_code, commune.nom_commune)  
  
SELECT  
dep_code AS "Département",  
nom_commune AS "Commune",  
round("valeur",1) AS "Prix moyen"  
FROM (  
SELECT dep_code, nom_commune, "valeur",  
rank() OVER (PARTITION BY dep_code ORDER BY valeur DESC) AS rang  
FROM valeur_par_ville) AS result  
WHERE rang <= 3;
```

Département	Commune	Prix moyen
06	Saint-Jean-Cap-Ferrat	968 750,00
06	Eze	655 000,00
06	Mouans-Sartoux	476 898,00
13	Gignac-la-Nerthe	330 000,00
13	Saint-Savournin	314 425,00
13	Cassis	313 416,90
33	Lège-Cap-Ferret	549 500,60
33	Vayres	335 000,00
33	Arcachon	307 435,90
59	Bersée	433 202,00
59	Cysoing	408 550,00
59	Halluin	322 250,00
69	Ville-sur-Jarnioux	485 300,00
69	Lyon 2e Arrondissement	455 217,30
69	Lyon 6e Arrondissement	426 968,30

R12- Les 20 communes avec le plus de transactions pour 1000 habitants pour les communes qui dépassent les 10 000 habitants

-- 12)Les 20 communes avec le plus de transactions
-- pour 1000 habitants pour les communes qui dépassent 10 000 habitants

SELECT

```
code_postal,  
nom_commune AS "nom commune",  
pop_totale AS "Population",
```

-- On arrondit la surface moyenne uniquement pour les lignes qui ont une surface

```
ROUND(AVG(NULLIF(surface_carrez, 0)), 0) AS "Surface Carrez moyenne",  
ROUND(AVG(valeur_mutation), 0) AS "Valeur foncière moyenne",
```

-- Ratio basé sur TOUTES les transactions de la table bien pour ces communes

```
ROUND((COUNT(id_bien) * 1000.0) / pop_totale, 2) AS "Nombre d'achat pour mille habitants"
```

FROM vente

JOIN bien USING (id_bien)

JOIN commune USING (code_dep_com)

WHERE pop_totale > 10000

GROUP BY nom_commune, code_postal, pop_totale

ORDER BY "Nombre d'achat pour mille habitants" DESC

LIMIT 20;

Code postal	Nom commune	Population	Surface Carrez moyenne	Valeur foncière moyenne	Nombre d'achat pour mille habitants
75002	Paris 2e Arrondissement	21 735	42	521 065	5,84
75001	Paris 1er Arrondissement	16 055	54	1 000 675	4,92
75003	Paris 3e Arrondissement	34 306	49	655 954	4,69
33120	Arcachon	11 898	50	307 436	4,62
44500	La Baule-Escoublac	16 797	50	264 745	4,58
75004	Paris 4e Arrondissement	29 390	48	645 485	4,08
06190	Roquebrune-Cap-Martin	13 041	47	274 173	3,99
75008	Paris 8e Arrondissement	36 250	75	1 010 536	3,83
83110	Sanary-sur-Mer	17 160	48	268 489	3,5
75009	Paris 9e Arrondissement	60 563	50	583 177	3,43
83250	La Londe-les-Maures	10 776	37	151 213	3,43
75006	Paris 6e Arrondissement	41 171	59	1 005 956	3,38
83270	Saint-Cyr-sur-Mer	11 725	53	256 894	3,24
60500	Chantilly	11 178	61	238 481	3,13
44380	Pornichet	11 440	44	204 127	3,06
94160	Saint-Mandé	22 576	55	494 061	3,06
75010	Paris 10e Arrondissement	86 863	49	532 282	3,04
06500	Menton	30 981	45	208 585	2,94
85270	Saint-Hilaire-de-Riez	11 501	36	85 265	2,87
94300	Vincennes	50 230	50	433 108	2,81

Merci !