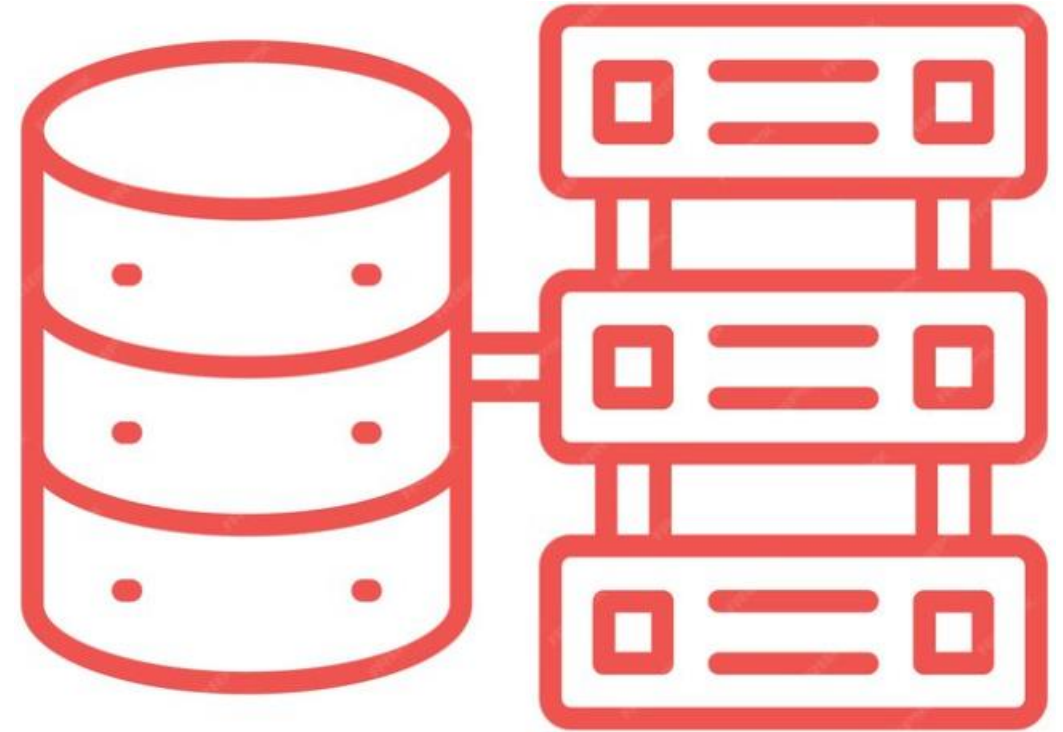


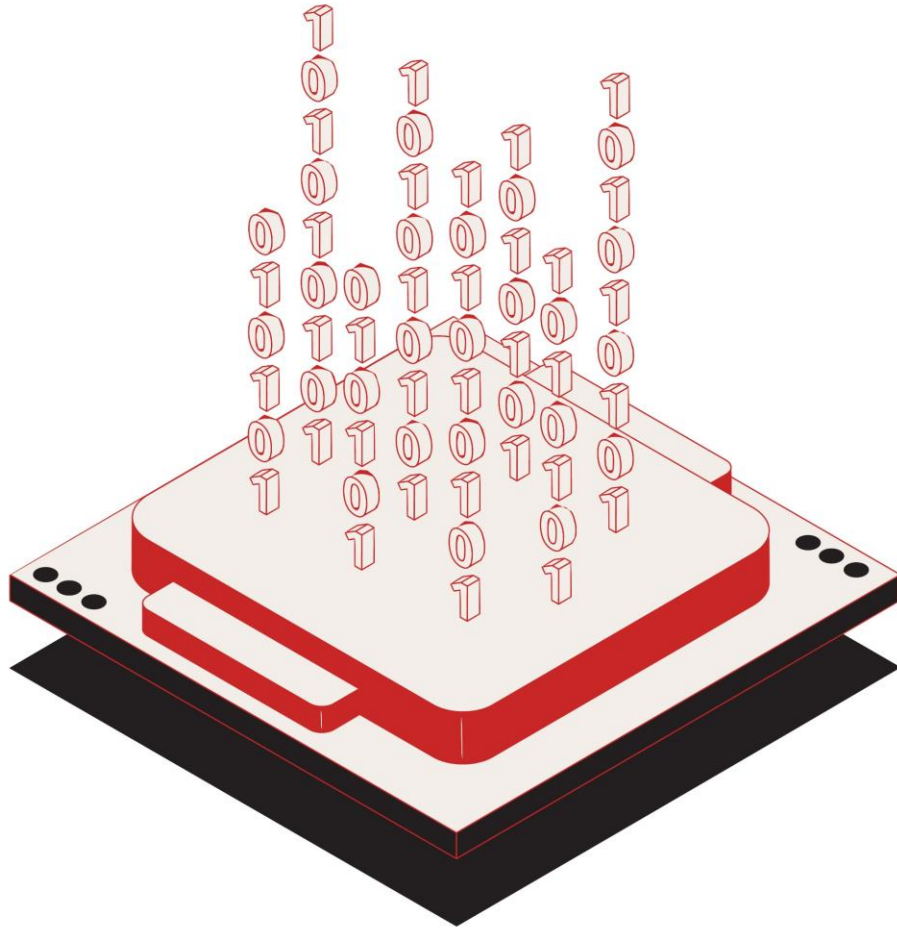


# Création et utilisation de la base de données



Le projet **DATAImmo** a pour but de regrouper et stocker les transactions des agences au sein d'une même base de données, afin de mieux prévoir le prix de vente des biens immobiliers et d'accompagner au mieux nos clients.





Afin d'être en  
conformité avec le  
**RGPD**, les données de  
ventes ont été  
anonymisées et celles-ci  
seront utilisés  
seulement pour les  
besoins de ce projet.



donnees\_communes

Recensement de la population par commune avec différentes informations sur la commune.

Source : **INSEE**



fr-esr-referentiel-geographique

données sur les régions, avec le référentiel géographique français, communes, unités urbaines, aires urbaines, départements, académies, régions

Source : **data.gouv**



Valeurs-foncières

Des données de demandes de valeurs foncières (DVF) comportant différentes informations sur les biens (prix, superficie, adresse, etc.)

Source : **data.gouv**





# L'extrait du dictionnaire des données

## DICTIONNAIRE DES DONNÉES - Adresse

CODE	SIGNIFICATION	TYPE	LONGUEUR	NATURE	REGLE DE GESTION	REGLE DE CALCUL
id_adresse	ID dans la base de données	Varchar	10	E	Ne doit pas être nul	
no_voie	Numéro de la rue	Varchar	5	E		
id_type_de_voie	Clé étrangère de la table Type_voie	Varchar	2	E	Ne doit pas être nul	
voie	Nom de la rue	Varchar	50	E	Ne doit pas être nul	
b_t_q	Indicateur de répétition pour l'adresse du logement	Char	1	E		
code_dep_code_commune	Clé étrangère de la table Commune	Varchar	6	CO	Ne doit pas être nul	

## DICTIONNAIRE DES DONNÉES - Bien

CODE	SIGNIFICATION	TYPE	LONGUEUR	NATURE	REGLE DE GESTION	REGLE DE CALCUL
id_bien	ID dans la base de données	Varchar	10	E	Ne doit pas être nul	
id_adresse	Clé étrangère de la table Adresse	Varchar	10	E	Ne doit pas être nul	
surface_carrez	Surface de l'habitation selon la loi Carrez	Float	NC	E	Ne doit pas être nul	
surface_reele	Surface de l'habitation	Integer	NC	E	Ne doit pas être nul	
id_type_local	Clé étrangère de la table Type_local	Varchar	2	E	Ne doit pas être nul	
nombre_pieces_principales	Nombre de pièces dans l'habitation	Varchar	2	E	Ne doit pas être nul	

## DICTIONNAIRE DES DONNÉES - Commune

CODE	SIGNIFICATION	TYPE	LONGUEUR	NATURE	REGLE DE GESTION	REGLE DE CALCUL
code_dep_code_commune	Fusion des codes dép et commune pour avoir une clé unique	Varchar	6	CO	Ne doit pas être nul	concaténation de code_dep et code_commune
code_dep	Clé étrangère de la table Département	Varchar	3	E	Ne doit pas être nul	
code_commune	Code de la commune	Varchar	3	E	Ne doit pas être nul	
commune	Libellé de la commune	Varchar	30	E	Ne doit pas être nul	
population	Population municipale de la commune	Integer	NC	E		ne prend pas en compte la population comptée à part

## DICTIONNAIRE DES DONNÉES - Département

CODE	SIGNIFICATION	TYPE	LONGUEUR	NATURE	REGLE DE GESTION	REGLE DE CALCUL
code_dep	Code du département	Varchar	3	E	Ne doit pas être nul	
departement	Libellé du département	Varchar	50	E	Ne doit pas être nul	
code_region	Clé étrangère de la table Région	Varchar	2	E	Ne doit pas être nul	

## DICTIONNAIRE DES DONNÉES - Région

CODE	SIGNIFICATION	TYPE	LONGUEUR	NATURE	REGLE DE GESTION	REGLE DE CALCUL
code_region	Code de la région	Varchar	2	E	Ne doit pas être nul	
region	Libellé de la région	Varchar	50	E	Ne doit pas être nul	

## DICTIONNAIRE DES DONNÉES - Type de local

CODE	SIGNIFICATION	TYPE	LONGUEUR	NATURE	REGLE DE GESTION	REGLE DE CALCUL
id_type_local	Code du type de local allant de 1 à 4 (1 ou 2 dans ce cas-ci)	Char	1	E	Ne doit pas être nul	
type_local	Type de local (Maison ou Appartement dans ce cas-ci)	Varchar	20	E	Ne doit pas être nul	

## DICTIONNAIRE DES DONNÉES - Type de voie

CODE	SIGNIFICATION	TYPE	LONGUEUR	NATURE	REGLE DE GESTION	REGLE DE CALCUL
id_type_voie	Nombre correspondant au type de la voie (rue, avenue, etc.)	Varchar	2	E	Ne doit pas être nul	
type_voie	Type de voie abrégé suivant la norme AFNOR NF Z10-011	Varchar	4	E		

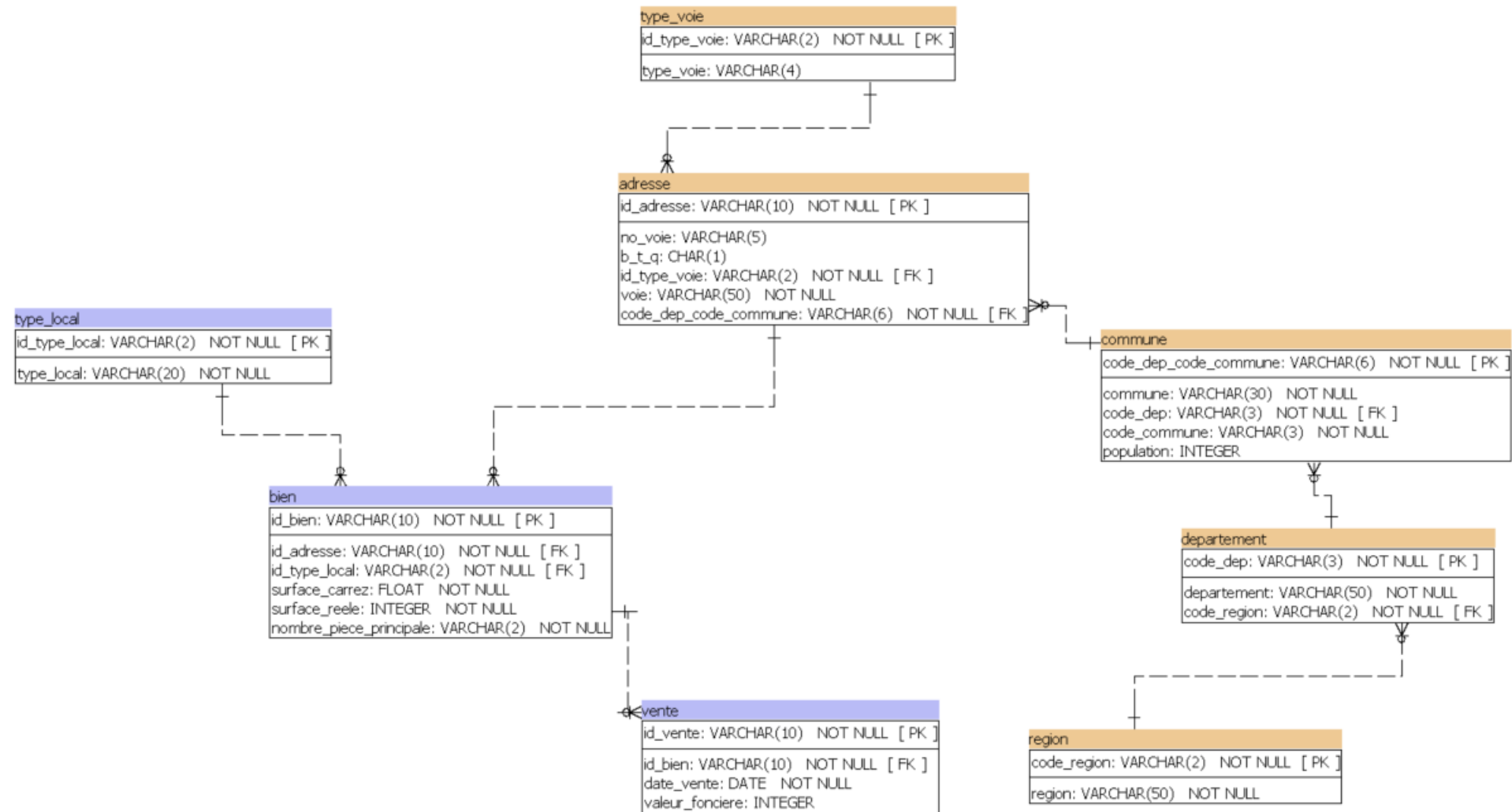
## DICTIONNAIRE DES DONNÉES - Vente

CODE	SIGNIFICATION	TYPE	LONGUEUR	NATURE	REGLE DE GESTION	REGLE DE CALCUL
id_vente	ID dans la base de données	Varchar	10	E	Ne doit pas être nul	
id_bien	Clé étrangère de la table Bien	Varchar	10	E	Ne doit pas être nul	
date_vente	Date de la signature de l'acte	Date	NC	E	aaaa-mm-jj	
valeur_fonciere	Valeur du bien	Integer	NC	E		net vendeur + TVA (hors frais de notaire)





# Le schéma relationnel normalisé





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

Structure Data Constraints Indexes Triggers DDL Total rows loaded: 34169

DATAImmo\_v2 Table name: Adresse WITHOUT ROWID STRICT

Name	Data type	Primary Key	Foreign Key	Unique	Check	Not NULL	Collate	Generated	Default value
1 id_adresse	VARCHAR (10)	Primary Key				Yes			NULL
2 no_voie	VARCHAR (5)					Yes			NULL
3 b_t_q	CHAR (1)					Yes			NULL
4 id_type_voie	VARCHAR (2)					Yes			NULL
5 voie	VARCHAR (50)					Yes			NULL
6 code_dep_code_commune	VARCHAR (6)					Yes			NULL

Structure Data Constraints Indexes Triggers DDL Total rows loaded: 109

DATAImmo\_v2 Table name: Departement WITHOUT ROWID STRICT

Name	Data type	Primary Key	Foreign Key	Unique	Check	Not NULL	Collate	Generated	Default value
1 code_dep	VARCHAR (3)	Primary Key				Yes			NULL
2 departement	VARCHAR (50)					Yes			NULL
3 code_region	VARCHAR (2)					Yes			NULL

Structure Data Constraints Indexes Triggers DDL Total rows loaded: 19

DATAImmo\_v2 Table name: Region WITHOUT ROWID STRICT

Name	Data type	Primary Key	Foreign Key	Unique	Check	Not NULL	Collate	Generated	Default value
1 code_region	VARCHAR (2)	Primary Key				Yes			NULL
2 region	VARCHAR (50)					Yes			NULL

Structure Data Constraints Indexes Triggers DDL Total rows loaded: 34169

DATAImmo\_v2 Table name: Bien WITHOUT ROWID STRICT

Name	Data type	Primary Key	Foreign Key	Unique	Check	Not NULL	Collate	Generated	Default value
1 id_bien	VARCHAR (10)	Primary Key				Yes			NULL
2 id_adresse	VARCHAR (10)					Yes			NULL
3 id_type_local	VARCHAR (2)					Yes			NULL
4 surface_carrez	FLOAT					Yes			NULL
5 surface_reele	INT					Yes			NULL
6 nombre_piece_principale	VARCHAR (2)					Yes			NULL

Structure Data Constraints Indexes Triggers DDL Total rows loaded: 2

DATAImmo\_v2 Table name: Type\_local WITHOUT ROWID STRICT

Name	Data type	Primary Key	Foreign Key	Unique	Check	Not NULL	Collate	Generated	Default value
1 id_type_local	VARCHAR (2)	Primary Key				Yes			NULL
2 type_local	VARCHAR (20)					Yes			NULL

Structure Data Constraints Indexes Triggers DDL Total rows loaded: 80

DATAImmo\_v2 Table name: Type\_voie WITHOUT ROWID STRICT

Name	Data type	Primary Key	Foreign Key	Unique	Check	Not NULL	Collate	Generated	Default value
1 id_type_voie	VARCHAR (2)	Primary Key				Yes			NULL
2 type_voie	VARCHAR (4)					Yes			NULL

Structure Data Constraints Indexes Triggers DDL Total rows loaded: 38916

DATAImmo\_v2 Table name: Commune WITHOUT ROWID STRICT

Name	Data type	Primary Key	Foreign Key	Unique	Check	Not NULL	Collate	Generated	Default value
1 code_dep_code_commune	VARCHAR (6)	Primary Key				Yes			NULL
2 commune	VARCHAR (30)					Yes			NULL
3 code_dep	VARCHAR (3)					Yes			NULL
4 code_commune	VARCHAR (3)					Yes			NULL
5 population	INT					Yes			NULL

Structure Data Constraints Indexes Triggers DDL Total rows loaded: 34169

DATAImmo\_v2 Table name: Vente WITHOUT ROWID STRICT

Name	Data type	Primary Key	Foreign Key	Unique	Check	Not NULL	Collate	Generated	Default value
1 id_vente	VARCHAR (10)	Primary Key				Yes			NULL
2 id_bien	VARCHAR (10)					Yes			NULL
3 date_vente	DATE					Yes			NULL
4 valeur_fonciere	FLOAT					Yes			NULL



# Requêtes SQL et résultats



# Requête 1 : Nombre total d'appartements vendus au 1er semestre 2020.



```
SELECT COUNT(Vente.id_vente) AS nb_vente_appt  
  
FROM Vente  
LEFT JOIN  
  Bien ON Vente.id_bien = Bien.id_bien  
  
WHERE id_type_local = 2
```

	nb_vente_appt
1	31378



## Requête 2 : Le nombre de ventes d'appartement par région pour le 1er semestre 2020.

```
SELECT COUNT(Vente.id_vente) AS nb_vente_appt,  
       Region.region,  
       Region.code_region  
  
FROM Vente  
  LEFT JOIN  
    Bien ON Vente.id_bien = Bien.id_bien  
  LEFT JOIN  
    Adresse ON Bien.id_adresse = Adresse.id_adresse  
  LEFT JOIN  
    Commune ON Adresse.code_dep_code_commune = Commune.code_dep_code_commune  
  LEFT JOIN  
    Departement ON Commune.code_dep = Departement.code_dep  
  LEFT JOIN  
    Region ON Departement.code_region = Region.code_region  
  
WHERE Bien.id_type_local = 2 AND  
       Vente.valeur_fonciere IS NOT NULL  
GROUP BY Region.region  
ORDER BY nb_vente_appt DESC
```

	nb_vente_appt	region	code_region
1	13989	Ile-de-France	11
2	3645	Provence-Alpes-Côte d'Azur	93
3	3253	Auvergne-Rhône-Alpes	84
4	1931	Nouvelle-Aquitaine	75
5	1640	Occitanie	76
6	1357	Pays de la Loire	52
7	1252	Hauts-de-France	32
8	984	Grand Est	44
9	983	Bretagne	53
10	861	Normandie	28
11	695	Centre-Val de Loire	24
12	376	Bourgogne-Franche-Comté	27
13	222	Corse	94
14	94	Martinique	02
15	44	La Réunion	04
16	34	Guyane	03
17	2	Guadeloupe	01



# Requête 3 : Proportion des ventes d'appartements par le nombre de pièces.

```
SELECT Bien.nombre_piece_principale AS nombre_de_piece,  
       ROUND(COUNT(Vente.id_vente) * 100.0 / total_ventes, 2) AS proportion_vente_appt  
  
FROM Vente  
  LEFT JOIN  
  Bien ON Vente.id_bien = Bien.id_bien  
  JOIN  
  (  
    SELECT COUNT(id_vente) AS total_ventes  
    FROM Vente  
      LEFT JOIN  
      Bien ON Vente.id_bien = Bien.id_bien  
    WHERE Bien.id_type_local = 2  
  )  
  
WHERE Bien.id_type_local = 2 AND  
       Vente.valeur_fonciere IS NOT NULL  
GROUP BY Bien.nombre_piece_principale  
ORDER BY nombre_de_piece
```

	nombre_de_piece	proportion_vente_appt
1	0	0.1
2	1	21.47
3	2	31.15
4	3	28.57
5	4	14.21
6	5	3.55
7	6	0.65
8	7	0.17
9	8	0.05
10	9	0.03
11	10	0.01
12	11	0



Requête 4 : Liste des 10 départements où le prix du mètre carré est le plus élevé.

```
SELECT ROUND(AVG(Vente.valeur_fonciere / Bien.surface_carrez), 0) AS prix_m2,  
       Departement.departement,  
       Departement.code_dep  
  
FROM Vente  
  JOIN  
    Bien ON Vente.id_bien = Bien.id_bien  
  JOIN  
    Adresse ON Bien.id_adresse = Adresse.id_adresse  
  JOIN  
    Commune ON Adresse.code_dep_code_commune = Commune.code_dep_code_commune  
  JOIN  
    Departement ON Departement.code_dep = Commune.code_dep  
  
WHERE Vente.valeur_fonciere IS NOT NULL  
GROUP BY Departement.departement  
ORDER BY prix_m2 DESC  
LIMIT 10
```

	prix_m2	Departement	Code_dep
1	12053	Paris	75
2	7219	Hauts-de-Seine	92
3	5343	Val-de-Marne	94
4	4700	Alpes-Maritimes	06
5	4667	Haute-Savoie	74
6	4345	Seine-Saint-Denis	93
7	4225	Yvelines	78
8	4059	Rhône	69
9	4027	Corse-du-Sud	2A
10	3764	Gironde	33



# Requête 5 : Prix moyen du mètre carré d'une maison en Île-de-France.

```
SELECT ROUND(AVG(Vente.valeur_fonciere / Bien.surface_carrez), 0) AS prix_moyen_m2_maison_idf

FROM Vente
  LEFT JOIN
    Bien ON Vente.id_bien = Bien.id_bien
  LEFT JOIN
    Adresse ON Bien.id_adresse = Adresse.id_adresse
  LEFT JOIN
    Commune ON Adresse.code_dep_code_commune = Commune.code_dep_code_commune
  LEFT JOIN
    Departement ON Commune.code_dep = Departement.code_dep

WHERE id_type_local = 1 AND
      code_region = 11 AND
      Vente.valeur_fonciere IS NOT NULL
```

	prix_moyen_m2_maison_idf
1	3745



# Requête 6 : Liste des 10 appartements les plus chers avec la région et le nombre de mètres carrés.

```
SELECT Vente.valeur_fonciere,  
       Region.region,  
       Bien.surface_carrez,  
       Bien.surface_reele  
  
FROM Vente  
      LEFT JOIN  
       Bien ON Vente.id_bien = Bien.id_bien  
      LEFT JOIN  
       Adresse ON Bien.id_adresse = Adresse.id_adresse  
      LEFT JOIN  
       Commune ON Adresse.code_dep_code_commune = Commune.code_dep_code_commune  
      LEFT JOIN  
       Departement ON Commune.code_dep = Departement.code_dep  
      LEFT JOIN  
       Region ON Departement.code_region = Region.code_region  
  
WHERE Bien.id_type_local = 2  
ORDER BY Vente.valeur_fonciere DESC  
LIMIT 10;
```

	valeur_fonciere	region	surface_carrez	surface_reele
1	9000000	Ile-de-France	9.1	10
2	8600000	Ile-de-France	64	62
3	8577713	Ile-de-France	20.55	289
4	7620000	Ile-de-France	42.77	42
5	7600000	Ile-de-France	253.3	200
6	7535000	Ile-de-France	139.9	143
7	7420000	Ile-de-France	360.95	357
8	7200000	Ile-de-France	595	241
9	7050000	Ile-de-France	122.56	310
10	6600000	Ile-de-France	79.38	76



# Requête 7 : Taux d'évolution du nombre de ventes entre le premier et le second trimestre de 2020.

```
WITH T1 AS (  
  SELECT COUNT(Vente.id_vente) AS nb_vente  
  FROM Vente  
  WHERE Vente.date_vente BETWEEN '2020-01-02' AND '2020-03-31'  
),  
  
T2 AS (  
  SELECT COUNT(Vente.id_vente) AS nb_vente  
  FROM Vente  
  WHERE Vente.date_vente BETWEEN '2020-04-01' AND '2020-06-30'  
)  
  
SELECT ROUND( ( (T2.nb_vente - T1.nb_vente) * 100.0) / T1.Nb_vente, 2) AS taux_evolution  
  
FROM T1,  
      T2
```

	taux_evolution
1	3.68



# Requête 8 : Le classement des régions par rapport au prix au mètre carré des appartement de plus de 4 pièces.

```
SELECT ROUND(AVG(Vente.valeur_fonciere / Bien.surface_carrez), 0) AS prix_m2,  
       Region.region,  
       Bien.id_type_local  
  
FROM Vente  
  LEFT JOIN  
    Bien ON Vente.id_bien = Bien.id_bien  
  LEFT JOIN  
    Adresse ON Bien.id_adresse = Adresse.id_adresse  
  LEFT JOIN  
    Commune ON Adresse.code_dep_code_commune = Commune.code_dep_code_commune  
  LEFT JOIN  
    Departement ON Commune.code_dep = Departement.code_dep  
  LEFT JOIN  
    Region ON Departement.code_region = Region.code_region  
  
WHERE Bien.nombre_piece_principale > 4 AND  
       Bien.id_type_local = 2 AND  
       Vente.valeur_fonciere IS NOT NULL  
  
GROUP BY Region.region  
ORDER BY prix_m2 DESC
```

	prix_m2	Region	id_type_local
1	8770	Ile-de-France	2
2	3642	La Réunion	2
3	3588	Provence-Alpes-Côte d'Azur	2
4	3105	Corse	2
5	2891	Auvergne-Rhône-Alpes	2
6	2465	Nouvelle-Aquitaine	2
7	2412	Bretagne	2
8	2316	Pays de la Loire	2
9	2190	Hauts-de-France	2
10	2097	Occitanie	2
11	2016	Normandie	2
12	1541	Grand Est	2
13	1453	Centre-Val de Loire	2
14	1251	Bourgogne-Franche-Comté	2
15	573	Martinique	2





# Requête 9 : Liste des communes ayant eu au moins 50 ventes au 1er trimestre.

```
SELECT COUNT(Vente.id_vente) AS nb_vente,  
       Commune.commune  
  
FROM Vente  
  LEFT JOIN  
    Bien ON Vente.id_bien = Bien.id_bien  
  LEFT JOIN  
    Adresse ON Bien.id_adresse = Adresse.id_adresse  
  LEFT JOIN  
    Commune ON Adresse.code_dep_code_commune = Commune.code_dep_code_commune  
  
WHERE Vente.date_vente BETWEEN '2020-01-02' AND '2020-03-31'  
GROUP BY Commune.commune  
  
HAVING nb_vente > 50  
ORDER BY nb_vente
```

	Nb_vente	Commune
1	53	Puteaux
2	54	Ajaccio
3	54	Versailles
4	56	Saint-Maur-des-Fossés
5	59	Levallois-Perret
6	59	Toulon
7	60	Paris 4e
8	61	Paris 2e
9	61	Rennes
10	62	La Ciotat
11	62	Paris 8e
12	62	Sète
13	63	Nîmes
14	64	Angers
15	65	Montreuil
16	66	Marseille 9e
17	67	Lille
18	68	Rueil-Malmaison
19	68	Vincennes
20	71	Marseille 1er
21	72	Marseille 4e
22	77	Antibes
23	78	Toulouse
24	79	Paris 3e
25	79	Paris 5e
26	80	Courbevoie
27	81	Asnières-sur-Seine
28	81	Marseille 8e
29	86	Paris 6e
30	87	Paris 7e
31	94	Paris 13e
32	99	Boulogne-Billancourt
33	106	Grenoble
34	106	Paris 9e
35	109	Paris 10e
36	110	Paris 12e
37	116	Paris 19e
38	119	Nantes
39	127	Paris 20e
40	146	Paris 14e
41	157	Bordeaux
42	165	Paris 16e
43	169	Paris 11e
44	173	Nice
45	209	Paris 18e
46	215	Paris 15e
47	228	Paris 17e



# Requête 10 : Différence en pourcentage du prix au mètre carré entre un appartement de 2 pièces et un appartement de 3 pièces.

```
WITH T2 AS (  
  SELECT ROUND(AVG(Vente.valeur_fonciere / Bien.surface_carrez), 2) AS prix_moyen_m2_appt_2_pcs  
  FROM Vente  
    JOIN  
      Bien ON Bien.id_bien = Vente.id_bien  
  WHERE Bien.nombre_piece_principale = 2 AND  
         Vente.valeur_fonciere IS NOT NULL AND  
         Bien.id_type_local = 2  
,  
  
T3 AS (  
  SELECT ROUND(AVG(Vente.valeur_fonciere / Bien.surface_carrez), 2) AS prix_moyen_m2_appt_3_pcs  
  FROM Vente  
    JOIN  
      Bien ON Bien.id_bien = Vente.id_bien  
  WHERE Bien.nombre_piece_principale = 3 AND  
         Vente.valeur_fonciere IS NOT NULL AND  
         Bien.id_type_local = 2  
)  
  
SELECT ROUND( ( (T3.prix_moyen_m2_appt_3_pcs - T2.prix_moyen_m2_appt_2_pcs) * 100.0) / T2.prix_moyen_m2_appt_2_pcs, 2) AS Diff_appt_2_vs_3  
  
FROM T2,  
      T3
```

	Diff_appt_2_vs_3
1	-12.4



# Requête11 : Les moyennes de valeurs foncières pour le top 3 des communes des départements 6, 13, 33, 59 et 69

```
SELECT valeur_fonciere,  
       commune,  
       code_dep  
  
FROM (  
  SELECT ROUND(AVG(Vente.valeur_fonciere), 0) AS valeur_fonciere,  
         Commune.commune,  
         Commune.code_dep,  
         ROW_NUMBER() OVER (PARTITION BY Commune.code_dep ORDER BY AVG(Vente.valeur_fonciere) DESC) AS row_num  
  FROM Vente  
  JOIN  
    Bien ON Vente.id_bien = Bien.id_bien  
  JOIN  
    Adresse ON Bien.id_adresse = Adresse.id_adresse  
  JOIN  
    Commune ON Adresse.code_dep_code_commune = Commune.code_dep_code_commune  
  WHERE Commune.code_dep IN ('06', '13', '33', '59', '69')  
  GROUP BY Commune.commune,  
           Commune.code_dep  
)  
AS ranked_sales  
  
WHERE row_num <= 3  
ORDER BY code_dep,  
       valeur_fonciere DESC
```

	valeur_fonciere	commune	code_dep
1	968750	Saint-Jean-Cap-Ferrat	06
2	655000	Eze	06
3	476898	Mouans-Sartoux	06
4	330000	Gignac-la-Nerthe	13
5	314425	Saint-Savournin	13
6	313417	Cassis	13
7	549501	Lège-Cap-Ferret	33
8	335000	Vayres	33
9	307436	Arcachon	33
10	433202	Bersée	59
11	408550	Cysoing	59
12	322250	Halluin	59
13	485300	Ville-sur-Jarnioux	69
14	455217	Lyon 2e	69
15	426968	Lyon 6e	69



# Requête 12 : Les 20 communes avec le plus de transactions pour 1000 habitants pour les communes qui dépassent les 10 000 habitants.

```
SELECT Commune.commune,  
       Commune.code_dep,  
       Commune.population,  
       ROUND( (COUNT(Vente.id_vente) * 1000.0) / Commune.population, 2) AS nb_vente_pour_1000  
FROM Vente  
LEFT JOIN  
  Bien ON Vente.id_bien = Bien.id_bien  
LEFT JOIN  
  Adresse ON Bien.id_adresse = Adresse.id_adresse  
LEFT JOIN  
  Commune ON Adresse.code_dep_code_commune = Commune.code_dep_code_commune  
WHERE Commune.population > 10000 AND  
       Vente.valeur_fonciere IS NOT NULL AND  
       Commune.population IS NOT NULL  
GROUP BY Commune.commune  
ORDER BY nb_vente_pour_1000 DESC  
LIMIT 20
```

	commune	code_dep	population	nb_vente_pour_1000
1	Paris 2e	75	21595	5.88
2	Paris 1er	75	15917	4.9
3	La Baule-Escoublac	44	16255	4.74
4	Paris 3e	75	34025	4.73
5	Arcachon	33	11630	4.73
6	Paris 4e	75	29131	4.08
7	Roquebrune-Cap-Martin	06	12839	4.05
8	Paris 8e	75	35655	3.9
9	Sanary-sur-Mer	83	16889	3.55
10	La Londe-les-Maures	83	10641	3.48
11	Paris 9e	75	60026	3.47
12	Paris 6e	75	40303	3.45
13	Saint-Cyr-sur-Mer	83	11484	3.22
14	Chantilly	60	10950	3.2
15	Pornichet	44	11141	3.14
16	Saint-Mandé	94	22377	3.08
17	Paris 10e	75	86472	3.05
18	Menton	06	30525	2.98
19	Saint-Hilaire-de-Riez	85	11297	2.92
20	Vincennes	94	49788	2.83



Merci