



Création et utilisation de la base de données

Rémi Bardey



Laplace Immo

Contexte du projet

Projet DATAImmo :

- **Laplace Immo** modernise sa base immobilière nationale pour se démarquer de la concurrence et mieux prévoir le prix de vente des biens immobiliers.
- **Mission analyste** : nettoyer, enrichir et **refondre le schéma 3 NF** + assurer conformité RGPD, puis créer un BDD.
- **Livrable clé** : base de données permettant de collecter les transactions immobilières et foncières en France.



La stratégie de sauvegarde et la conformité RGPD



Conformité RGPD

- Champs “Nom” & “Prénom” supprimés des tables VENTE et BIEN dès la création.
- Seuls identifiants techniques (ID vente, ID bien) et données utiles à l’analyse conservées.



Sauvegarde des requêtes

- Notebook DATAImmo.ipynb → export des requêtes SQL validées avec commentaires.
- Copies versionnées sur Drive sécurisé (accès restreint, historique)
- Résultat : base conforme RGPD et requêtes pérennisées pour le suivi du projet.

Les données initiales

Fichier CSV	Source open data	Volume indicatif	Contenu clé
valeursfoncieres.csv	DVF (data.gouv)	filtré sur l'année 2020	prix, date, type de bien, surface, code commune
referentiel_geo.csv	Référentiel géographique (data.gouv)	19 régions 109 départements	Codes INSEE + hiérarchie territoriale
communes_insee.csv	INSEE – recensement	+34 k communes	population, densité, indicateurs socio-démographiques

L'extrait du dictionnaire des données

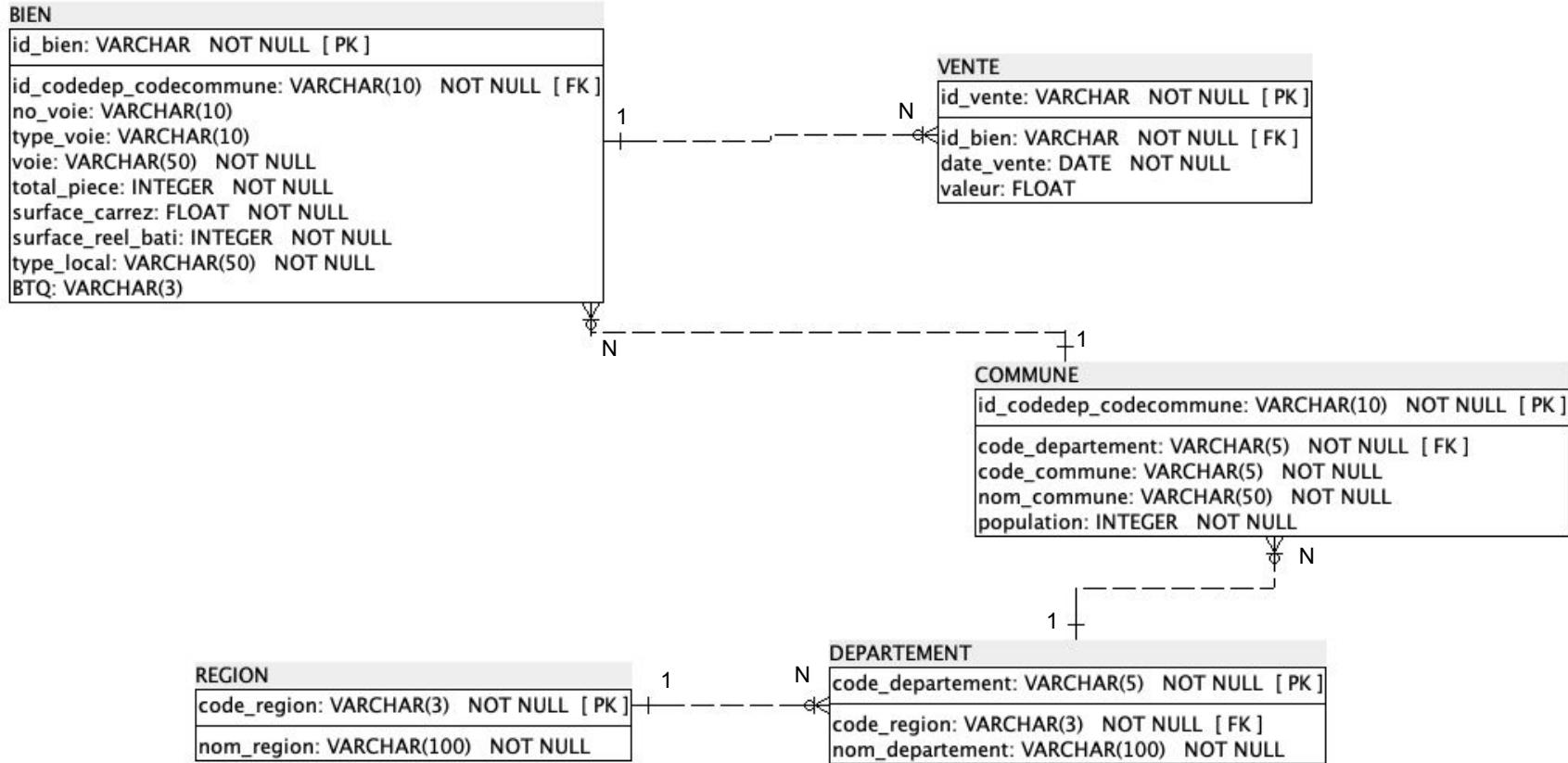
Table BIEN

CODE	SIGNIFICATION	TYPE	LONGUEUR	NATURE	REGLE DE GESTION	REGLE DE CALCUL
id_bien	clé primaire	VARCHAR	n/c	élémentaire	Auto-incrémenté - not null	
id_codedep_codecommune	clé étrangère	VARCHAR	10	concaténation	not null	code département + code commune
no_voie	numéro de voie du bien	VARCHAR	10	élémentaire	nullable	
type_voie	rue, avenue, allée etc...	VARCHAR	10	élémentaire	nullable	
voie	nom de la voie	VARCHAR	50	élémentaire	not null	
total_piece	Nombre de pièce d'un bien	INTEGER	n/c	élémentaire	not null	
surface_carrez	surface carrez du bien	FLOAT	n/c	élémentaire	not null	
type_local	appartement ou maison	VARCHAR	50	élémentaire	not null	
BTQ	referenciel postal	VARCHAR	3	élémentaire	nullable	

Table COMMUNE

CODE	SIGNIFICATION	TYPE	LONGUEUR	NATURE	REGLE DE GESTION	REGLE DE CALCUL
id_codedep_codecommune	clé primaire	VARCHAR	10	concaténation	code département + code commune - not null	
code_departement	clé étrangère	VARCHAR	5	élémentaire	format 2 chiffres - not null	
code_commune	code du département	VARCHAR	5	élémentaire	format 3 chiffres - not null	
nom_commune	nom de la commune	VARCHAR	50	élémentaire	not null	
population	population total par commune	INTEGER	n/c	calculé	not null	résident permanent + résident non permanent

Le schéma relationnel normalisé



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

Tables créées dans SQLite

Tables (5)
> BIEN
> COMMUNE
> DEPARTEMENT
> REGION
> VENTE
Views

Exemple de données chargées : Table BIEN

id_bien	id_codedep_codecommune	no_voie	type_voie	voie	Total rows loaded: 34169			
					total_piece	surface_carrez	type_local	BTQ
1	01103	347	RUE	DU CHATEAU	3	48.22	Appartement	
2	06004	4	BD	EDOUARD BAUDOIN	1	39.11	Appartement	
3	06088	20	RUE	MARCEAU	3	80.25	Appartement	B
4	06123	550	RTE	DES VESPINS RN7	1	27.51	Appartement	
5	13005	9300	RES	LES ARPEGES BD DES ... ^P^A	2	47.33	Appartement	
6	13028	27	RUE	DU GRAND MADIER	1	25.31	Appartement	

Les requêtes ou screenshot qui permettent de démontrer le bon chargement des données

```
1 SELECT
2 (SELECT COUNT(*) FROM REGION) AS nb_region,
3 (SELECT COUNT(*) FROM DEPARTEMENT) AS nb_departement,
4 (SELECT COUNT(*) FROM COMMUNE) AS nb_commune,
5 (SELECT COUNT(*) FROM BIEN) AS nb_bien,
6 (SELECT COUNT(*) FROM VENTE) AS nb_vente;
7
```

Total rows loaded: 1

	nb_region	nb_departement	nb_commune	nb_bien	nb_vente
1	19	109	34991	34169	34169



Requêtes SQL et résultats

Sous-titre

Requête 1

Nombre total d'appartements vendus au 1er semestre 2020

```
1 SELECT COUNT(b.id_bien) AS nombre_appartement_vendu_S1_2020
2 FROM BIEN b
3 INNER JOIN VENTE v ON b.id_bien = v.id_bien
4 WHERE b.type_local = 'Appartement'
5 AND v.date_vente
6 BETWEEN '2020-01-01' AND '2020-06-30'
```

The screenshot shows a MySQL Workbench interface with a dark theme. At the top, there is a toolbar with various icons. Below the toolbar, the status bar displays "Total rows loaded: 1". The main area contains the SQL query and its results.

	nombre_appartement_vendu_S1_2020
1	31377

Requête 2

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

```
1 SELECT
2     r.nom_region AS région,
3     COUNT(*) AS nb_ventes_appartement_S1_2020
4 FROM VENTE v
5     INNER JOIN BIEN b
6         ON v.id_bien = b.id_bien
7     INNER JOIN COMMUNE c
8         ON b.id_codedep_codecommune = c.id_codedep_codecommune
9     INNER JOIN DEPARTEMENT d
10        ON c.code_departement = d.code_departement
11     INNER JOIN REGION r
12        ON d.code_region = r.code_region
13 WHERE b.type_local = 'Appartement'
14     AND v.date_vente
15     BETWEEN '2020-01-01' AND '2020-06-30'
16 GROUP BY
17     r.nom_region
18 ORDER BY
19     nb_ventes_appartement_S1_2020 DESC
```

	région	nb_ventes_appartement_S1_2020
1	Ile-de-France	13995
2	Provence-Alpes-Côte d'Azur	3649
3	Auvergne-Rhône-Alpes	3253
4	Nouvelle-Aquitaine	1932
5	Occitanie	1640
6	Pays de la Loire	1356
7	Hauts-de-France	1254
8	Grand Est	984
9	Bretagne	983
10	Normandie	862
11	Centre-Val de Loire	696
12	Bourgogne-Franche-Comté	376
13	Corse	223
14	Martinique	94
15	La Réunion	44
16	Guyane	34
17	Guadeloupe	2

Requête 3

Proportion des ventes d'appartements par le nombre de pièces

```
1 SELECT
2     b.total_piece AS pieces,
3     COUNT(*) AS nb_ventes,
4     ROUND(100.0 * COUNT(*) / (
5         SELECT COUNT(*)
6             FROM VENTE v
7             INNER JOIN BIEN b
8                 ON v.id_bien = b.id_bien
9                 WHERE b.type_local = 'Appartement'
10            ), 2) AS pct_ventes
11
12 FROM VENTE v
13 INNER JOIN BIEN b
14     ON v.id_bien = b.id_bien
15 WHERE b.type_local = 'Appartement'
16 GROUP BY b.total_piece
17 ORDER BY b.total_piece;
18
```

	pieces	nb_ventes	pct_ventes
1	0	30	0.1
2	1	6739	21.48
3	2	9782	31.18
4	3	8966	28.58
5	4	4460	14.21
6	5	1114	3.55
7	6	204	0.65
8	7	54	0.17
9	8	17	0.05
10	9	8	0.03
11	10	2	0.01
12	11	1	0

Requête 4

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

```
1 SELECT
2   d.nom_departement AS departement,
3   ROUND(AVG(v.valeur / b.surface_carrez), 0) AS prix_m2
4 FROM VENTE v
5 INNER JOIN BIEN b
6   ON v.id_bien = b.id_bien
7 INNER JOIN COMMUNE c
8   ON b.id_codedep_codecommune = c.id_codedep_codecommune
9 INNER JOIN DEPARTEMENT d
10 ON c.code_departement = d.code_departement
11 WHERE b.type_local = 'Appartement'
12 AND b.surface_carrez > 0
13 GROUP BY
14   c.code_departement
15 ORDER BY
16   prix_m2 DESC
17 LIMIT 10;
```

	departement	prix_m2
1	Paris	12045
2	Hauts-de-Seine	7194
3	Val-de-Marne	5424
4	Alpes-Maritimes	4683
5	Haute-Savoie	4654
6	Seine-Saint-Denis	4359
7	Yvelines	4259
8	Rhône	4083
9	Corse-du-Sud	3883
10	Hautes-Alpes	3798

Requête 5

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

```
1 SELECT
2 ROUND(AVG(v.valeur / b.surface_carrez), 0) AS Prix_moyen_maison_IDF
3 FROM VENTE v
4 INNER JOIN BIEN b ON v.id_bien = b.id_bien
5 INNER JOIN COMMUNE c ON b.id_codedep_codecommune = c.id_codedep_codecommune
6 INNER JOIN DEPARTEMENT d ON c.code_departement = d.code_departement
7 INNER JOIN REGION r ON d.code_region = r.code_region
8 WHERE UPPER(TRIM(r.nom_region)) = 'ILE-DE-FRANCE'
9 AND b.type_local = 'Maison'
10 AND b.surface_carrez > 0 ;|
```

The screenshot shows a MySQL command-line interface window. At the top, there are several icons: a blue square with a white question mark, a green checkmark, a red X, and four blue arrows pointing left, right, up, and down. To the right of these are buttons for executing the current statement (blue arrow right), redoing the last statement (blue arrow right with a dot), and exiting (red X). Next is a page number button labeled '1'. Following these are standard MySQL navigation keys: a blue arrow left, a blue arrow right, a blue arrow up, and a blue arrow down. To the right of these keys is a small icon of a document with a printer symbol. To the right of the document icon is the text 'Total rows loaded: 1'. Below this, there is a table structure with two columns. The first column is labeled 'Prix_moyen_maison_IDF' and the second column is labeled '3745'. A horizontal line separates this from another row where the first column is labeled '1' and the second column is also labeled '3745'.

Prix_moyen_maison_IDF	3745
1	3745

Requête 6

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

```
1 SELECT
2 v.id_bien AS id_bien,
3 v.valeur AS Prix,
4 b.surface_carrez AS Surface,
5 r.nom_region AS Region
6 FROM VENTE v
7 INNER JOIN BIEN b ON v.id_bien = b.id_bien
8 INNER JOIN COMMUNE c ON b.id_codedep_codecommune = c.id_codedep_codecommune
9 INNER JOIN DEPARTEMENT d ON c.code_departement = d.code_departement
10 INNER JOIN REGION r ON d.code_region = r.code_region
11 WHERE b.type_local = 'Appartement'
12 AND v.valeur > ''
13 ORDER BY valeur DESC
14 LIMIT 10;
```

	id_bien	Prix	Surface	Region
1	30603	9000000.0	9.1	Ile-de-France
2	5261	8600000	64	Ile-de-France
3	3624	8577713	20.55	Ile-de-France
4	7601	7620000	42.77	Ile-de-France
5	9988	7600000	253.3	Ile-de-France
6	17823	7535000	139.9	Ile-de-France
7	410	7420000	360.95	Ile-de-France
8	16357	7200000	595	Ile-de-France
9	1924	7050000	122.56	Ile-de-France
10	19161	6600000	79.38	Ile-de-France

Requête 7

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

```
1 WITH
2   ventes_t1 AS (
3     SELECT COUNT(*) AS nb_t1
4     FROM VENTE
5     WHERE date_vente BETWEEN '2020-01-01' AND '2020-03-31') ,
6
7 ventes_t2 AS (
8   SELECT COUNT(*) AS nb_t2
9   FROM VENTE
10  WHERE date_vente BETWEEN '2020-04-01' AND '2020-06-30')
11
12 SELECT
13   ROUND((ventes_t2.nb_t2 - ventes_t1.nb_t1) * 100.0 / ventes_t1.nb_t1, 2)
14   AS taux_evolution
15 FROM ventes_t1, ventes_t2;
```

	taux_evolution
1	3.68

Requête 8

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

```
1 SELECT
2   r.nom_region AS Region,
3   ROUND(AVG(v.valeur / b.surface_carrez), 2) AS prix_m2
4 FROM VENTE v
5 INNER JOIN BIEN b
6   ON v.id_bien = b.id_bien
7 INNER JOIN COMMUNE c
8   ON b.id_codedep_codecommune = c.id_codedep_codecommune
9 INNER JOIN DEPARTEMENT d
10  ON c.code_departement = d.code_departement
11 INNER JOIN REGION r
12  ON d.code_region = r.code_region
13 WHERE b.type_local = 'Appartement'
14  AND b.total_piece > 4
15  AND v.valeur IS NOT NULL
16  AND b.surface_carrez > 0
17 GROUP BY r.nom_region
18 ORDER BY prix_m2 DESC;
```

	Region	prix_m2
1	Ile-de-France	8757.64
2	La Réunion	3641.81
3	Provence-Alpes-Côte d'Azur	3587.65
4	Corse	3104.88
5	Auvergne-Rhône-Alpes	2891.38
6	Nouvelle-Aquitaine	2465.48
7	Bretagne	2412.05
8	Pays de la Loire	2315.76
9	Hauts-de-France	2189.93
10	Occitanie	2097.23
11	Normandie	2015.77
12	Grand Est	1540.89
13	Centre-Val de Loire	1453.11
14	Bourgogne-Franche-Comté	1251.19
15	Martinique	573.48

Requête 9

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

```
1 WITH ventes_t1 AS (
2     SELECT
3         b.id_codedep_codecommune AS code_insee,
4         c.nom_commune           AS commune,
5         COUNT(*)                 AS nb_ventes
6     FROM VENTE v
7     INNER JOIN BIEN b
8         ON v.id_bien = b.id_bien
9     INNER JOIN COMMUNE c
10        ON b.id_codedep_codecommune = c.id_codedep_codecommune
11    WHERE v.date_vente BETWEEN '2020-01-01' AND '2020-03-31'
12    GROUP BY b.id_codedep_codecommune, c.nom_commune
13 )
14 SELECT
15     commune,
16     nb_ventes
17 FROM ventes_t1
18 WHERE nb_ventes >= 50
19 ORDER BY nb_ventes DESC;
```

	Commune	nb_ventes
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
8	Paris 14e Arrondissement	146
9	Paris 20e Arrondissement	127
10	Nantes	119
11	Paris 19e Arrondissement	116
12	Paris 12e Arrondissement	110
13	Paris 10e Arrondissement	109
14	Paris 9e Arrondissement	106
15	Grenoble	106
16	Boulogne-Billancourt	99

Requête 10

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

```
1 WITH stats AS (
2     SELECT
3         b.total_piece AS pieces,
4         COUNT(*) AS nb_ventes,
5         AVG(v.valeur / b.surface_carrez) AS prix_m2
6     FROM VENTE v
7     INNER JOIN BIEN b
8         ON v.id_bien = b.id_bien
9     WHERE b.type_local = 'Appartement'
10        AND v.valeur IS NOT NULL
11        AND b.surface_carrez > 0
12        AND b.total_piece IN (2, 3)
13     GROUP BY b.total_piece
14 )
15 SELECT
16     s2.nb_ventes AS nb_ventes_2,
17     s2.prix_m2 AS prix_m2_2,
18     s3.nb_ventes AS nb_ventes_3,
19     s3.prix_m2 AS prix_m2_3,
20     ROUND( 100.0 * (s3.prix_m2 - s2.prix_m2) / s2.prix_m2 , 2) AS difference_pct
21 FROM stats s2
22 INNER JOIN stats s3
23 ON s2.pieces = 2
24 AND s3.pieces = 3;
25 |
```

	nb_ventes_2	prix_m2_2	nb_ventes_3	prix_m2_3	difference_pct
1	9782	4903.7146076091785	8966	4299.883118292375	-12.31

Requête 11

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

```
1 WITH sales_commune AS (
2   SELECT
3     c.code_departement,
4     c.nom_commune,
5     COUNT(*)          AS nb_ventes,
6     AVG(v.valeur)    AS avg_valeur
7   FROM VENTE v
8   INNER JOIN BIEN b
9     ON v.id_bien = b.id_bien
10  INNER JOIN COMMUNE c
11    ON b.id_codedep_codecommune = c.id_codedep_codecommune
12  WHERE c.code_departement IN ('06', '13', '33', '59', '69')
13    AND v.valeur IS NOT NULL
14  GROUP BY
15    c.code_departement,
16    c.nom_commune
17 ),
18
19 ranked_communes AS (
20   SELECT
21     code_departement,
22     nom_commune,
23     nb_ventes,
24     avg_valeur,
25     ROW_NUMBER() OVER (
26       PARTITION BY code_departement
27       ORDER BY nb_ventes DESC
28     ) AS rn
29   FROM sales_commune
30 )
31
32 SELECT
33   code_departement,
34   nom_commune AS commune,
35   ROUND(avg_valeur, 2) AS moyenne_valeur_fonciere
36 FROM ranked_communes
37 WHERE rn <= 3
38 ORDER BY
39   code_departement,
40   rn;
```

	code_departement	commune	moyenne_valeur_fonciere
1	06	Nice	273715.95
2	06	Antibes	256005.46
3	06	Menton	208584.88
4	13	Marseille 8e Arrondissement	209837.13
5	13	Marseille 4e Arrondissement	115943.62
6	13	Marseille 1er Arrondissement	160203.93
7	33	Bordeaux	253232.17
8	33	Mérignac	208139.92
9	33	Talence	183914.72
10	59	Lille	206779.14
11	59	Roubaix	144011.81
12	59	La Madeleine	200915.94
13	69	Villeurbanne	196082.8
14	69	Lyon 9e Arrondissement	227282.6
15	69	Lyon 3e Arrondissement	347104.21

Requête 12

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

```
1 WITH tx_per_commune AS (
2   SELECT
3     c.nom_commune AS commune,
4     c.population,
5     COUNT(*) AS nb_transactions
6   FROM VENTE v
7   INNER JOIN BIEN b
8     ON v.id_bien = b.id_bien
9   INNER JOIN COMMUNE c
10    ON b.id_codedep_codecommune = c.id_codedep_codecommune
11   WHERE v.valeur IS NOT NULL
12     AND c.population > 10000
13   GROUP BY
14     c.nom_commune,
15     c.population)
16 SELECT
17   commune,
18   nb_transactions,
19   population,
20   ROUND(nb_transactions * 1000.0 / population, 2) AS tx_transactions_pour_1000
21 FROM tx_per_commune
22 ORDER BY tx_transactions_pour_1000 DESC
23 LIMIT 20;
```

Ajouter un pied de page

	commune	nb_transactions	population	tx_transactions_pour_1000
1	Paris 2e Arrondissement	127	21735	5.84
2	Paris 1er Arrondissement	79	16055	4.92
3	Paris 3e Arrondissement	161	34306	4.69
4	Arcachon	55	11898	4.62
5	La Baule-Escoublac	77	16797	4.58
6	Paris 4e Arrondissement	120	29390	4.08
7	Roquebrune-Cap-Martin	52	13041	3.99
8	Paris 8e Arrondissement	139	36250	3.83
9	Sanary-sur-Mer	60	17160	3.5
10	La Londe-les-Maures	37	10776	3.43
11	Paris 9e Arrondissement	208	60563	3.43
12	Paris 6e Arrondissement	139	41171	3.38
13	Saint-Cyr-sur-Mer	38	11725	3.24
14	Chantilly	35	11178	3.13
15	Pornichet	35	11440	3.06
16	Saint-Mandé	69	22576	3.06
17	Paris 10e Arrondissement	264	86863	3.04
18	Menton	91	30981	2.94
19	Saint-Hilaire-de-Riez	33	11501	2.87
20	Vincennes	141	50230	2.81

21

Conclusion

Base “DATAImmo” refondue : schéma 3 NF, dictionnaire et tables implantés (SQLite)

Données intégrées & nettoyées : DVF + INSEE + référentiel géographique, sans nom/prénom ⇒ conformité RGPD

Traçabilité assurée : notebook versionné → requêtes validées & sauvegardées sur Drive sécurisé

Socle prêt pour la suite : analyses régionales & **modèle de prédiction des prix** pour toutes les agences Laplace Immo



Merci !