

# POC implémentation base de données immobilières

---

*Données du premier semestre 2020*

## 01. Dictionnaire des données :

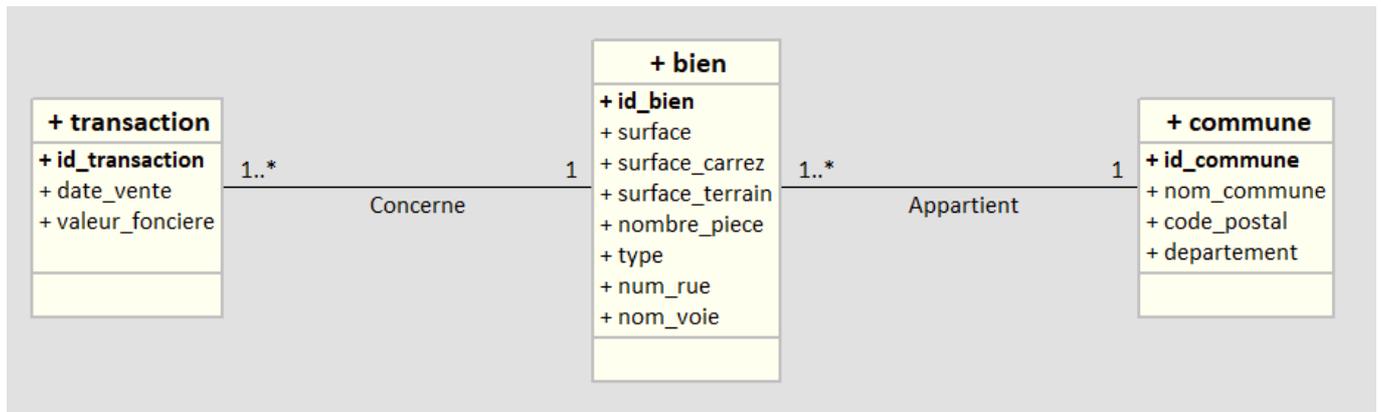
Il faut tout d'abord extraire de notre source de données les éléments nécessaires à notre analyse. J'ai ici fait le choix de conserver les données essentielle à nos futures analyse afin d'éviter de surcharger la base avec des informations inutiles ou redondantes.

Dictionnaire des données :

Numéro	Code propriété	Signification	Type	Observation
1	id_bien	Identifiant du bien	Numérique	Identifiant
2	id_transaction	Identifiant de la transaction	Numérique	Identifiant
3	id_commune	Identifiant de la commune	Numérique	Identifiant
4	date_vente	Date de vente	Date	
5	valeur_fonciere	Prix de vente	Monétaire	
6	surface	Surface du bien	Numérique	
7	surface_carrez	Surface Carrez du bien	Numérique	
8	surface_terrain	Surface du terrain	Numérique	
9	nombre_piece	Nombre de pièces principales	Numérique	
10	type	Type du bien	Texte	Longueur : 15
11	num_rue	Numéro de rue	Alphanumérique	Longueur : 15
12	nom_voie	Rue de L'article	Texte	
13	nom_commune	Ville de l'article	Texte	
14	code_postal	Code Postal de l'article	Numérique	
15	departement	Departement du bien	Alphanumérique	

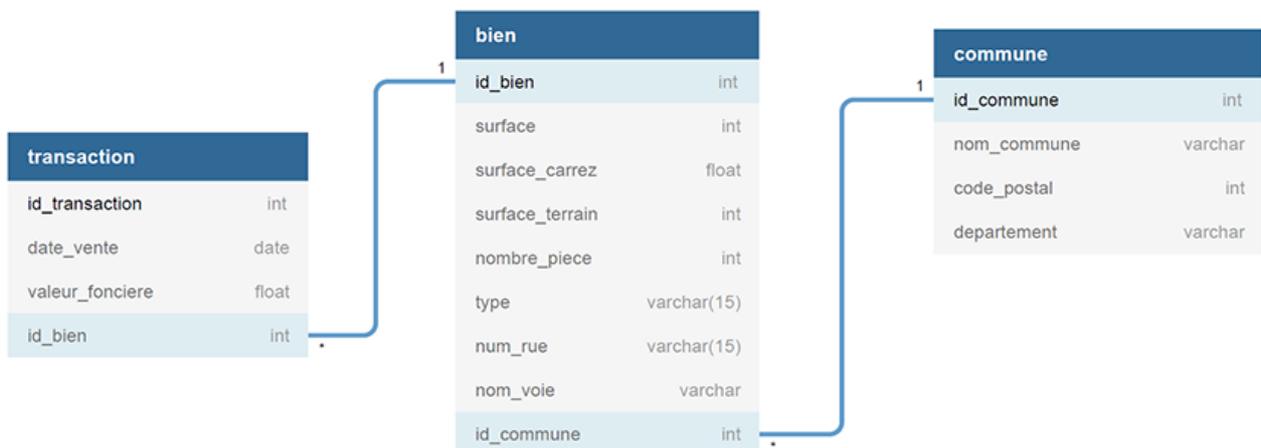
## 02. Modèle Conceptuel des données :

Modélisation au formalisme UML de la structure de nos données :



## 03. Schéma relationnel normalisé en 3NF

Ressort de cette préparation un schéma relationnel prêt pour l'implémentation tel que :



## 04. Implémentation de la base de donnée

Une fois nos tables créées et importées dans une base PostgreSQL, nous pouvons alors interroger la base :

Table commune :

	123 id_commune	ABC nom_commune	123 code_postal	ABC departement
1	0	SAINT-ETIENNE-DU-BOIS	1 370	1
2	1	CHEVRY	1 170	1
3	2	DIVONNE-LES-BAINS	1 220	1
4	3	PERON	1 630	1
5	4	VALSERHONE	1 200	1
6	5	CULOZ	1 350	1
7	6	ST-GENIS-POUILLY	1 630	1
8	7	OYONNAX	1 100	1
9	8	ST-GERMAIN-DE-JOUX	1 130	1
10	9	GEX	1 170	1
11	10	AMBERIEU-EN-BUGEY	1 500	1
12	11	CESSY	1 170	1

Table bien :

	123 id_bien	123 surface	123 surface_carrez	123 surface_terrain	123 nombre_piece	ABC type	ABC num_rue	ABC nom_voie	123 id_commune
1	1	52	50,42	[NULL]	2	Appartement	190 A	RUE CENTRALE	0
2	2	48	48,22	[NULL]	3	Appartement	347	RUE DU CHATEAU	1
3	3	130	130,8	[NULL]	6	Appartement	58	AV DU MONT BLANC	2
4	4	109	109,22	[NULL]	5	Maison	140	RUE DE L'ABBE JOLIVET	3
5	5	91	108,65	[NULL]	4	Appartement	39	RUE BUFFON	4
6	6	32	31,65	[NULL]	2	Appartement	28	AV JEAN FALCONNIER	5
7	7	52	52,58	[NULL]	2	Appartement	8	RUE DE GENEVE	6
8	8	60	58,71	[NULL]	2	Appartement	2	RUE DU RECULET	6
9	9	96	93,23	[NULL]	4	Maison	1403	RUE JEAN DE GINGINS	2
10	10	117	117	[NULL]	5	Maison	226	ALL DES CAPUCINES	6
11	11	36	35,6	[NULL]	2	Appartement	276	RTE DE POUGNY	3
12	12	137	138,03	[NULL]	5	Appartement	79	CRS DE VERDUN	7

Table transaction :

	123 id_transaction	🕒 date_vente	123 valeur_fonciere	123 id_bien
1	1	2020-02-03	56 000	1
2	2	2020-01-02	165 000	2
3	3	2020-01-08	720 000	3
4	4	2020-01-06	429 250	4
5	5	2020-01-07	220 900	5
6	6	2020-01-21	42 000	6
7	7	2020-01-07	262 000	7
8	8	2020-01-08	190 000	8
9	9	2020-01-16	563 130	9
10	10	2020-01-17	535 000	10
11	11	2020-01-16	330 000	11
12	12	2020-01-27	110 600	12

## 05. Interrogation de la base :

Passons à l'exécution des 9 requêtes demandées.

R\_01 : Nombre total d'appartements vendus au 1er semestre 2020.

Requête :

```
SELECT count(DISTINCT transaction.id_bien)
FROM TRANSACTION
NATURAL JOIN bien
WHERE bien."type" = 'Appartement';
```

Résultat : 31 378

R\_02 : Proportion des ventes d'appartements par le nombre de pièces.

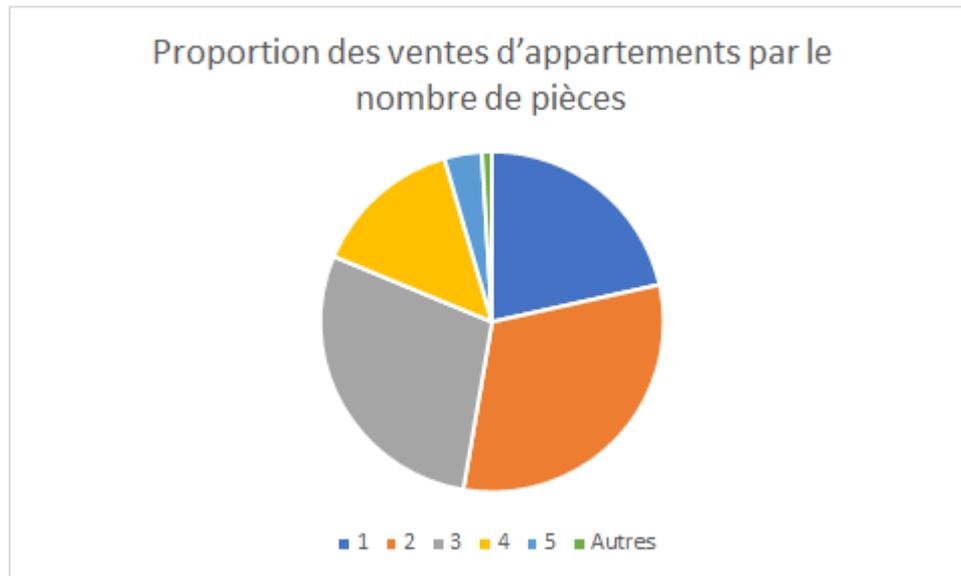
Requête :

```
SELECT bien.nombre_piece,
       round((count(bien."type")::decimal * 100)/
            (SELECT count(bien.type)
             FROM bien
             WHERE bien."type" = 'Appartement')::decimal, 3) AS pourcentage
FROM TRANSACTION
NATURAL JOIN bien
WHERE bien."type" = 'Appartement'
GROUP BY bien.nombre_piece
ORDER BY bien.nombre_piece ;
```

Résultats :

<b>nombre_piece</b>	<b>pourcentage</b>
0	0.096
1	21.491
2	31.198
3	28.592
4	14.223
5	3.553
6	0.651
7	0.172

nombre_piece	pourcentage
8	0.054
9	0.026
10	0.006
11	0.003



R\_03 : Liste des 10 départements où le prix du mètre carré est le plus élevé.

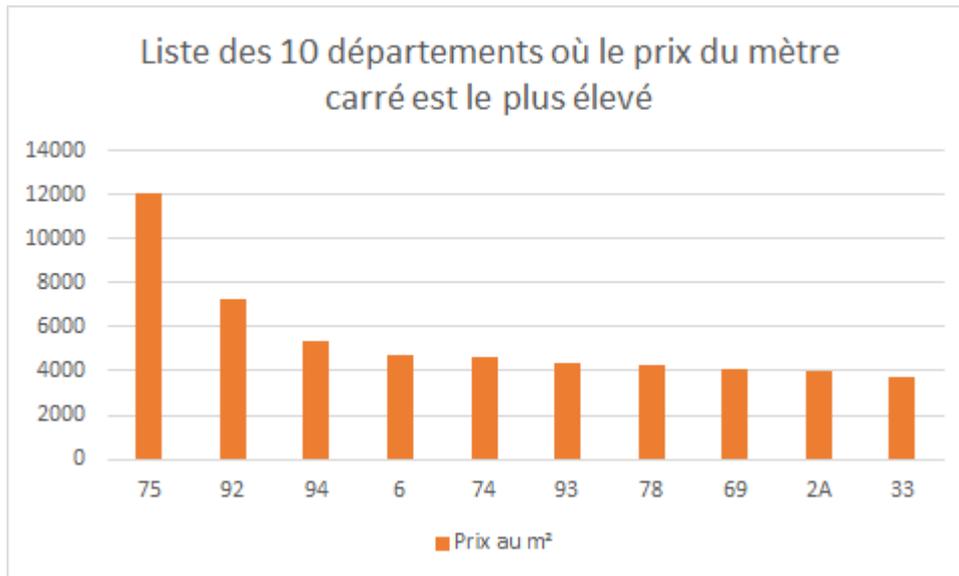
Requête :

```
SELECT departement,
       round(avg(transaction.valeur_fonciere / bien.surface_carrez)::decimal, 2)
AS prix_au_m2
FROM bien
NATURAL JOIN commune
NATURAL JOIN TRANSACTION
GROUP BY departement
ORDER BY prix_au_m2 DESC
LIMIT 10;
```

Résultats :

departement	prix_au_m2
75	12052.89
92	7219.39
94	5343.28
6	4700.33

departement	prix_au_m2
74	4667.13
93	4344.78
78	4225.25
69	4059.31
2A	4026.97
33	3764.14



R\_04 : Prix moyen du mètre carré d'une maison en Île-de-France.

Requête :

```
SELECT round(avg(valeur_fonciere/surface_carrez)::decimal, 2)
FROM commune
NATURAL JOIN TRANSACTION
NATURAL JOIN bien
WHERE TYPE = 'Maison'
AND departement in ('75',
                    '77',
                    '78',
                    '91',
                    '92',
                    '93',
                    '94',
                    '95') ;
```

Résultat : 3745.01

R\_05 : Liste des 10 appartements les plus chers avec le département et le nombre de mètres carrés.

Requête :

```
SELECT id_bien,
       valeur_fonciere,
       departement,
       surface_carrez
FROM TRANSACTION
NATURAL JOIN bien
NATURAL JOIN commune
WHERE bien.type = 'Appartement'
      AND valeur_fonciere IS NOT NULL
ORDER BY valeur_fonciere DESC
LIMIT 10;
```

Résultats :

id_bien	valeur_fonciere	departement	surface_carrez
32252	9000000.0	75	9.1
21817	8600000.0	91	64.0
29778	8577713.0	75	20.55
32410	7620000.0	75	42.77
29829	7600000.0	75	253.3
29501	7535000.0	75	139.9
31950	7420000.0	75	360.95
32112	7200000.0	75	595.0
29332	7050000.0	75	122.56
29492	6600000.0	75	79.38

R\_06 : Taux d'évolution du nombre de ventes entre le premier et le second trimestre de 2020.

Requête :

```
WITH nombre_vente_t1 AS
  (SELECT count(transaction.date_vente) AS t1
   FROM TRANSACTION
   WHERE date_vente <= '2020-03-31'),
  nombre_vente_t2 AS
  (SELECT count(transaction.date_vente) AS t2
   FROM TRANSACTION
   WHERE date_vente >= '2020-04-01')
SELECT round(((t2 - t1)::decimal / t1)*100, 2) AS taux_evolution
FROM nombre_vente_t1
CROSS JOIN nombre_vente_t2
```

Résultat : 3.68

R\_07 : Liste des communes où le nombre de ventes a augmenté d'au moins 20% entre le premier et le second trimestre de 2020.

Requête :

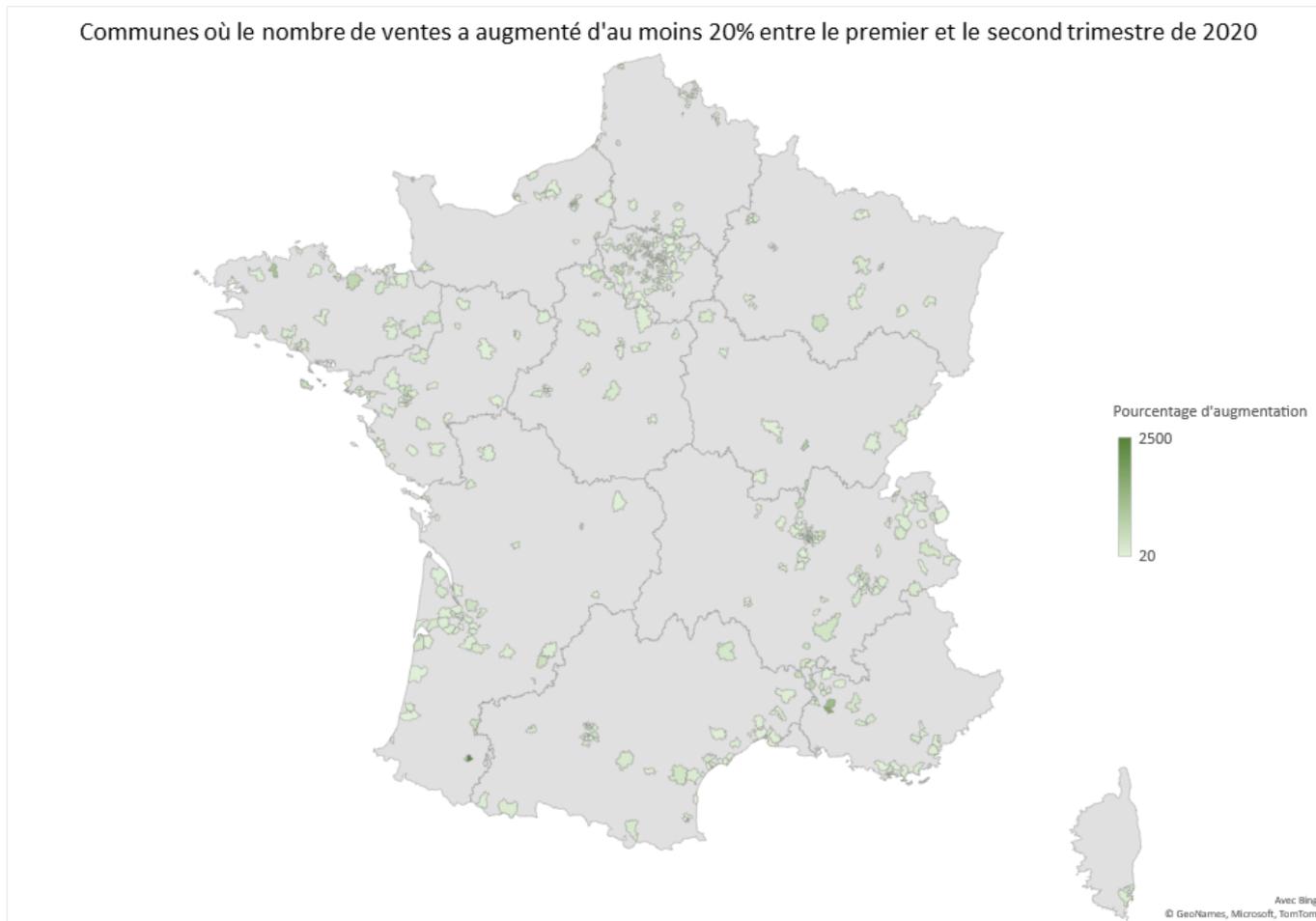
```
WITH nombre_vente_t1 AS
  (SELECT count(transaction.id_transaction) AS t1,
    nom_commune
  FROM TRANSACTION
  NATURAL JOIN bien
  NATURAL JOIN commune
  WHERE date_vente <= '2020-03-31'
  GROUP BY nom_commune),
  nombre_vente_t2 AS
  (SELECT count(transaction.id_transaction) AS t2,
    nom_commune
  FROM TRANSACTION
  NATURAL JOIN bien
  NATURAL JOIN commune
  WHERE date_vente >= '2020-04-01'
  GROUP BY nom_commune)
SELECT nom_commune,
  round(((t2 - t1)::decimal / t1) * 100, 2) AS taux_evolution
FROM nombre_vente_t1
NATURAL JOIN nombre_vente_t2
WHERE ((t2 - t1)::decimal / t1) * 100 >= 20
ORDER BY taux_evolution DESC
```

Résultats :

nom_commune	taux_evolution
PAU	2500.00
CAVAILLON	1600.00
L ISLE SUR LA SORGUE	1200.00
MORLAIX	1000.00
LE DEVOLUY	1000.00
...	...
GOLBEY	20.00
ARCACHON	20.00
LE BEAUSSET	20.00

nom_commune	taux_evolution
SAINTE FOY LES LYON	20.00
DIVONNE-LES-BAINS	20.00

574 lignes retournées



R\_08 : Différence en pourcentage du prix au mètre carré entre un appartement de 2 pièces et un appartement de 3 pièces.

Requête :

```
WITH prix_au_m2_2 AS
  (SELECT round(avg(valeur_fonciere/surface_carrez)::decimal, 2) AS prix_2
   FROM TRANSACTION
   NATURAL JOIN bien
   WHERE nombre_piece = 2
         AND TYPE = 'Appartement'),
  prix_au_m2_3 AS
  (SELECT round(avg(valeur_fonciere/surface_carrez)::decimal, 2) AS prix_3
   FROM TRANSACTION
   NATURAL JOIN bien
   WHERE nombre_piece = 3
         AND TYPE = 'Appartement')
SELECT round(((prix_3 - prix_2)::decimal / prix_2) * 100, 2) AS taux_evolution
```

```
FROM prix_au_m2_2
CROSS JOIN prix_au_m2_3
```

Résultat : -12.40

R\_09 : Les moyennes de valeurs foncières pour le top 3 des communes des départements 6, 13, 33, 59 et 69.

Requête :

```
SELECT *
FROM
  (SELECT nom_commune,
    departement,
    round(avg(valeur_fonciere)::decimal, 2) AS valeur_moyenne,
    row_number() OVER (PARTITION BY departement
      ORDER BY avg(valeur_fonciere) DESC) AS classement
  FROM TRANSACTION
  NATURAL JOIN bien
  NATURAL JOIN commune
  WHERE departement in ('6',
    '33',
    '13',
    '59',
    '69'))
  GROUP BY nom_commune,
    departement) AS classement
WHERE classement <= 3 ;
```

Résultats :

nom_commune	departement	valeur_moyenne	classement
GIGNAC-LA-NERTHE	13	330000.00	1
SAINT SAVOURNIN	13	314425.00	2
CASSIS	13	313416.88	3
LEGE-CAP-FERRET	33	549500.64	1
VAYRES	33	335000.00	2
ARCACHON	33	307435.93	3
BERSEE	59	433202.00	1
CYSOING	59	408550.00	2
HALLUIN	59	322250.00	3
SAINT-JEAN-CAP-FERRAT	6	968750.00	1

<b>nom_commune</b>	<b>departement</b>	<b>valeur_moyenne</b>	<b>classement</b>
EZE	6	655000.00	2
MOUANS-SARTOUX	6	476898.10	3
VILLE SUR JARNIOUX	69	485300.00	1
LYON 2EME	69	455217.27	2
LYON 6EME	69	426968.25	3

## 06. Conclusion

Notre implémentation des données nous permet un travail précis d'analyse tout en nous permettant d'étendre son utilisation à un jeu de données actualisé afin de réaliser des études sur les données en cours.