

DATAImmo

Projet Immobilier



Laplace Immo



Contexte et Besoins

L'agence immobilière « **Laplace immo** » nous a confié la [création d'une base de données](#), dans laquelle est collectée les transactions immobilières et foncières en France.

L'objectif est l'utilisation de celle-ci pour [analyser le marché](#) et par conséquent venir en aide aux différentes agences dans [l'accompagnement](#) de leurs clients.



Table des matières

Solution 01

Création de la base 02
de données

Analyse des données 03



SOLUTION

La première étape :

- Analyse du marché & des données
- Sélection des données

La seconde étape :

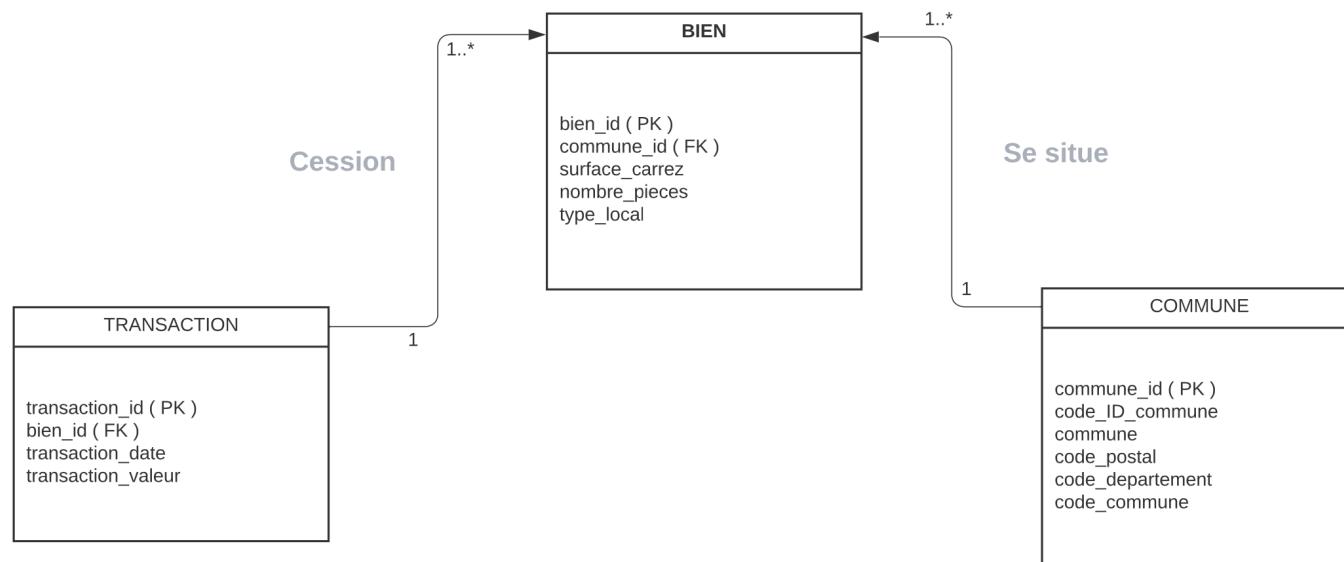
- Création du model conceptuel de données
- Réalisation du schéma relationnel normalisé en
BNF

Dictionnaire des données

Code mnémonique	Désignation	Type	Taille	Remarque TRANSACTIONS
bien_id	Identification du bien	AN	30	Formule : CONCAT (code de département /code de commune / Préfixe de section / Section / Numéro de plan /1er lot)
transaction_date	Date de la mutation du bien	Date	10	Au format MM/DD/YYYY
transaction_valeur	Valeure foncière	N	15	prix de la transaction en euro €
transaction_id	identitification de la transaction	AN	30	
commune_id	identification de la commune	AN	30	Formule : CONCAT (code de département /code de commune / code ID de commune)
Code_departement	code du département	AN	5	
Code_commune	code de la commune	N	5	
bien1_lot	1 er lot	N	5	
Nombre_lots	nombre de lots	N	2	> 0 , NOT NULL
Type_local	type de local	A	15	
Surface_carrez	surface carrez du lot	N	2	> 0 , NOT NULL
Nombre_pieces	nombre de pièce lot	N	2	> 0 , NOT NULL
Code_ID_commune	code ID de la commune	N	5	
Code_postal	code postal	N	5	
Commune	commune	A	25	

Model conceptuel des données UML

DIAGRAMME UML



Model conceptuel des données UML

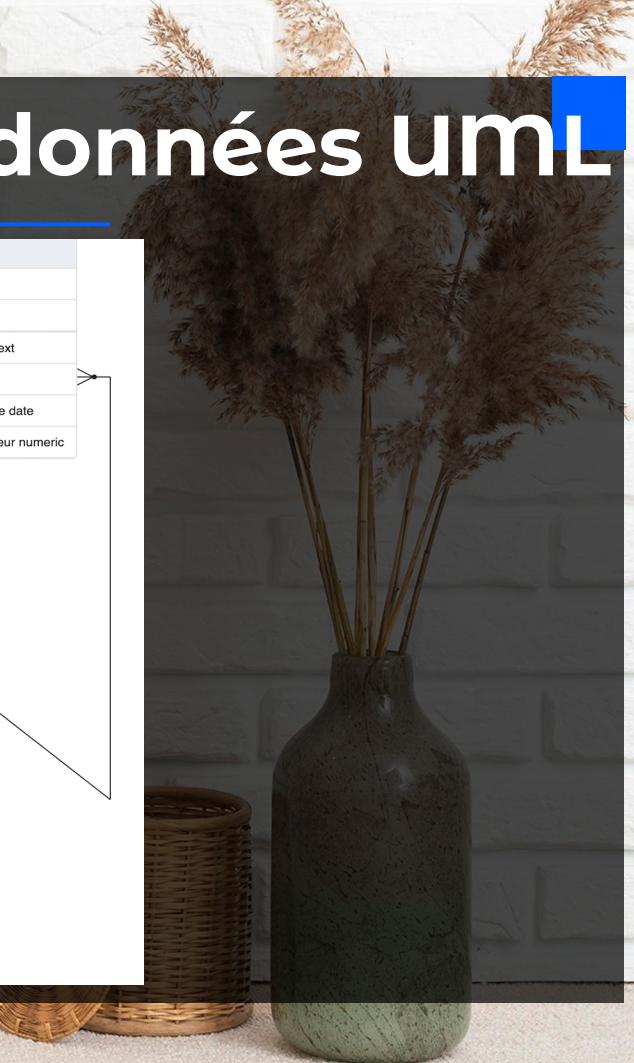
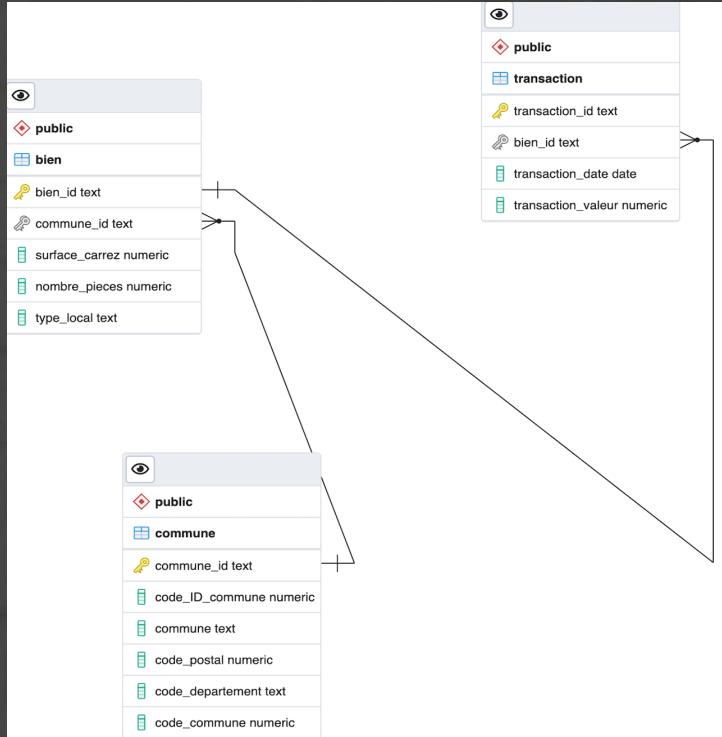


Schéma relationnel normalisé

➤ BIEN

bien_id, commune_id, surface_carrez,
nombre_pieces, type_local

➤ commune

commune_id, code_ID_commune,
commune, code_postal,
code_departement, code_commune

➤ TRANSACTION

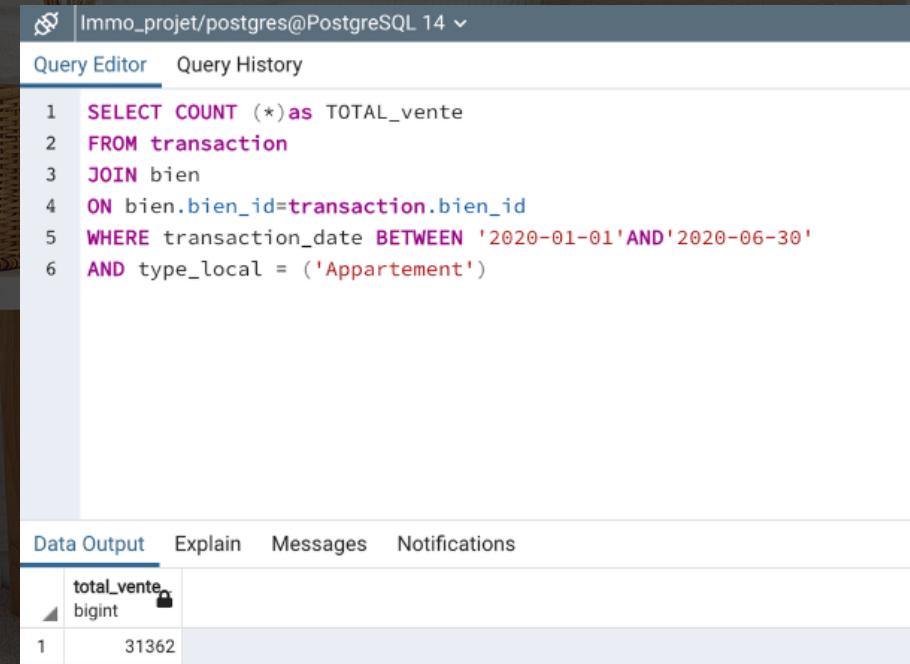
Transaction_id, bien_id, transaction_date,
transaction_valeur

ANALYSE DES DONNÉES

- n°1 : Nombre total d'appartements vendus au 1er semestre 2020

Requête

Résultat



The screenshot shows a PostgreSQL query editor interface. The title bar reads "Immo_projet/postgres@PostgreSQL 14". The main area is the "Query Editor" tab, which contains the following SQL code:

```
1 SELECT COUNT (*)as TOTAL_vente
2 FROM transaction
3 JOIN bien
4 ON bien.bien_id=transaction.bien_id
5 WHERE transaction_date BETWEEN '2020-01-01'AND'2020-06-30'
6 AND type_local = ('Appartement')
```

Below the code, there are tabs for "Data Output", "Explain", "Messages", and "Notifications". The "Data Output" tab is selected and displays the following result table:

	total_vente
1	31362

ANALYSE DES DONNÉES

➤ N°2 : Proportion des ventes d'appartements par le nombre de pièces

Requête

```
Query Editor Query History
1 SELECT ROUND ( SUM (nombre_pieces)/(SELECT SUM (nombre_pieces)FROM bien)*100,2)
2 AS proportion_ventes_pieces ,nombre_pieces,
3 SUM(nombre_pieces) AS total_des_ventes FROM bien
4 WHERE type_local ='Appartement'
5 GROUP by nombre_pieces
6 ORDER BY nombre_pieces
```

Résultat

	proportion_ventes_pieces numeric	nombre_pieces numeric	total_des_ventes numeric
1	0.00	0	0
2	7.54	1	6737
3	21.89	2	19564
4	30.09	3	26892
5	19.96	4	17836
6	6.23	5	5570
7	1.37	6	1224

ANALYSE DES DONNÉES

- N°3 : Liste des 10 départements où le prix du mètre carré est le + élevé

Requête

```
1 SELECT ROUND(SUM(transaction_valeur)/SUM(surface_carrez))AS prix_moyen, code_departement
2 FROM transaction
3 JOIN bien USING (bien_id)
4 JOIN commune USING (commune_id)
5 GROUP BY code_departement
6 ORDER BY prix_moyen DESC
7 LIMIT 10
```

Data Output Explain Messages Notifications

	prix_moyen numeric	code_departement text
1	11912	75
2	7238	92
3	4833	94
4	4591	6
5	4159	74
6	4074	93
7	4002	78
8	3894	69
9	3765	2A
10	3568	33

Résultat



ANALYSE DES DONNÉES

- n°4 : Prix moyen du mètre carré d'une maison en île-de-France

Requête

```
1 SELECT ROUND (AVG (transaction_valeur/surface_carrez),2)AS prix_M2_maison
2 FROM transaction
3 JOIN bien USING (bien_id)
4 JOIN commune USING (commune_id)
5 WHERE code_departement IN ('75','78','77','95','94','93','92','91')AND type_local ='Maison'
```

Résultat

prix_m2_maison	numeric
1	3745.01

ANALYSE DES DONNÉES

- N°5 : Liste des 10 appartements les plus cher avec le département et le nombre de mètres carrés

Query Editor Query History

```
1 SELECT transaction_valeur, code_departement, surface_carrez
2 FROM transaction
3 JOIN bien USING (bien_id)
4 JOIN commune USING (commune_id)
5 WHERE type_local = 'Appartement'
6 ORDER BY transaction_valeur DESC
7 LIMIT 10
```

Requête

Data Output				Explain	Messages	Notifications
	transaction_valeur numeric	code_departement text	surface_carrez numeric			
1	9000000.00	75			9.1	
2	8600000.00	91			64	
3	8577713.00	75			20.55	
4	7620000.00	75			42.77	
5	7600000.00	75			253.3	
6	7535000.00	75			139.9	
7	7420000.00	75			360.95	
8	7200000.00	75			595	
9	7050000.00	75			122.56	
10	6600000.00	75			79.38	

Résultat

ANALYSE DES DONNÉES

- N°6 : Taux d'évolution du nombre de ventes entre le premier et le second trimestre de 2020

```
1 WITH
2 tab1 AS
3 (SELECT COUNT (bien_id) AS vente_1trimestre
4 FROM transaction
5 WHERE transaction_date BETWEEN '2020-01-01' AND '2020-03-31'),
6
7 tab2 AS
8 (SELECT COUNT (bien_id) AS vente_2trimestre
9 FROM transaction
10 WHERE transaction_date BETWEEN '2020-04-01' AND '2020-06-30')
11
12 SELECT ROUND(((vente_2trimestre - vente_1trimestre )*100.0)/(vente_1trimestre ),2) AS taux_evolution
13 FROM tab1, tab2
14
```

Requête

	taux_evolution
	numeric
1	3.66

Résultat

ANALYSE DES DONNÉES

- N°7 : Liste des communes où le nombre de vente a augmenté d'au moins 20% entre le premier et le second trimestre de 2020

```
1 WITH
2 tab1 AS (SELECT commune, COUNT(transaction_valeur)AS Val1 FROM transaction
3           JOIN bien ON bien.bien_id=transaction.bien_id
4           JOIN commune ON commune.commune_id=bien.commune_id
5           WHERE transaction_date BETWEEN '2020-01-01'AND '2020-03-30'
6           GROUP BY commune ),
7
8 tab2 AS(SELECT commune,COUNT(transaction_valeur)AS Va2 FROM transaction
9           JOIN bien ON bien.bien_id=transaction.bien_id
0           JOIN commune ON commune.commune_id=bien.commune_id
1           WHERE transaction_date BETWEEN '2020-04-01'AND '2020-06-30'
2           GROUP BY commune )
3
4           SELECT tab1.commune,ROUND((Va2-VA1)/(Va1)*100,2)AS taux_evolution FROM tab1
5           JOIN tab2 ON tab1.commune=tab2.commune
6           WHERE ROUND((Va2-VA1)/(Va1)*100,2)>20
7
```

Requête

Résultat

	commune	taux_evolution
1	CHANTONNAY	100.00
2	CHATEAUDUN	200.00
3	RETHEL	600.00
4	CREPY EN VALOIS	100.00
5	PONT ST ESPRIT	400.00
6	GOUSSAINVILLE	100.00
7	LA FERTE BERNARD	100.00
8	MORLAIX	1000.00

ANALYSE DES DONNÉES

- N°8 : Différence en pourcentage du prix au mètre carré entre un appartement de 2 pièces et un appartement de 3 pièces

```
1 WITH
2 prix_nbr2 AS
3 (SELECT AVG (transaction_valeur/surface_carrez)AS prix_m2_appart_2p
4 FROM transaction
5 JOIN bien ON bien.bien_id=transaction.bien_id
6 WHERE nombre_pieces='2'AND type_local = 'Appartement'AND surface_carrez IS NOT NULL AND transaction_valeur
7 IS NOT NULL),
8 prix_nbr3 AS
9 (SELECT AVG (transaction_valeur/surface_carrez)AS prix_m2_appart_3p
10 FROM transaction
11 JOIN bien ON bien.bien_id=transaction.bien_id
12 WHERE nombre_pieces='3'AND type_local = 'Appartement'AND surface_carrez IS NOT NULL AND transaction_valeur
13 IS NOT NULL)
14 SELECT ((prix_m2_appart_3p-prix_m2_appart_2p)/(prix_m2_appart_2p)*100) AS evolution_en_pourcentage
15 FROM prix_nbr2,prix_nbr3
```

Requête

Résultat

evolution_en_pourcentage
numeric
-12.40039521881570351300

ANALYSE DES DONNÉES

➤ N°9 : Les moyennes de valeurs foncières pour le top 3 des communes des départements 6, 13, 33, 59 et 69

Query Editor Query History

```
1 WITH
2 prix_moyen_communne AS
3 (SELECT commune, code_departement,(ROUND (AVG(transaction_valeur),2))AS prix_moyen
4 FROM transaction
5 JOIN bien USING (bien_id)
6 JOIN commune USING (commune_id)
7 GROUP BY commune, code_departement
8 ),
9 ranking_communne AS
10 (SELECT RANK ()OVER (
11     PARTITION BY code_departement ORDER BY prix_moyen DESC) AS ranking, commune, code_departement, prix_moyen
12 FROM prix_moyen_communne
13 WHERE code_departement IN ('6','13', '33','59','69'))
14
15 SELECT * FROM ranking_communne
16 WHERE ranking < 4
```

Requête

Résultat

	ranking bigint	commune text	code_departement text	prix_moyen numeric
1	1	GIGNAC-LA-NERTHE	13	330000.00
2	2	SAINT SAVOURNIN	13	314425.00
3	3	CASSIS	13	313416.88
4	1	LEGE-CAP-FERRET	33	549500.64
5	2	VAYRES	33	335000.00
6	3	ARCACHON	33	307435.93
7	1	BERSEE	59	433202.00
8	2	CYSOING	59	408550.00
9	3	HALLUIN	59	322250.00
10	1	SAINT-JEAN-CAP-FERRAT	6	968750.00
11	2	EZE	6	655000.00
12	3	MOUANS-SARTOUX	6	476898.10
13	1	VILLE SUR JARNIQUX	69	485300.00
14	2	LYON 2EME	69	455217.27
15	3	LYON 6EME	69	426968.25