

Analyse des données : (Requêtes SQL)

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

The screenshot shows a PostgreSQL database interface with the following details:

- Connection:** data_immo/postgres@PostgreSQL 18
- Toolbar:** Includes icons for file operations, search, and various database functions.
- Query History:** Shows the executed SQL script.
- Script Content:**

```
1 -- 1)Nombre total d'appartements vendus au 1er semestre 2020.
2
3 SELECT count(*) AS "Nb appart vendus"
4 FROM vente
5
6 JOIN bien ON bien.id_bien = vente.id_bien
7
8 WHERE bien.type_local = 'Appartement'
9 AND date_mutation BETWEEN '2020-01-01' AND '2020-06-30';
10
11
```
- Data Output:** A table showing the result of the query.

	Nb appart vendus
1	31378
- Bottom Bar:** Includes icons for file operations, search, and various database functions, along with pagination information: Showing rows: 1 to 1 / Page No: 1 / of 1.

Script SQL

```
-- 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';
```

Résultat

Nb appart vendus
S1-2020
31378

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

The screenshot shows a PostgreSQL client interface with the following details:

- Connection:** data_immo/postgres@PostgreSQL 18
- Query History:** Shows the executed SQL script.
- SQL Script:**

```
--2)Le nombre de ventes d'appartement par région pour le 1er semestre 2020.
SELECT region.reg_nom AS Région,
       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
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 "NB_ventes" DESC;
```
- Data Output:** A table showing the results of the query.

	région character varying (50)	NB_ventes bigint
1	Ile-de-France	13995
2	Provence-Alpes-Côte d'A...	3649
3	Auvergne-Rhône-Alpes	3253
4	Nouvelle-Aquitaine	1932

- Total rows:** 17
- Query complete:** 00:00:00.122
- Bottom right:** CRLF, Ln 1, Col 1

Script SQL

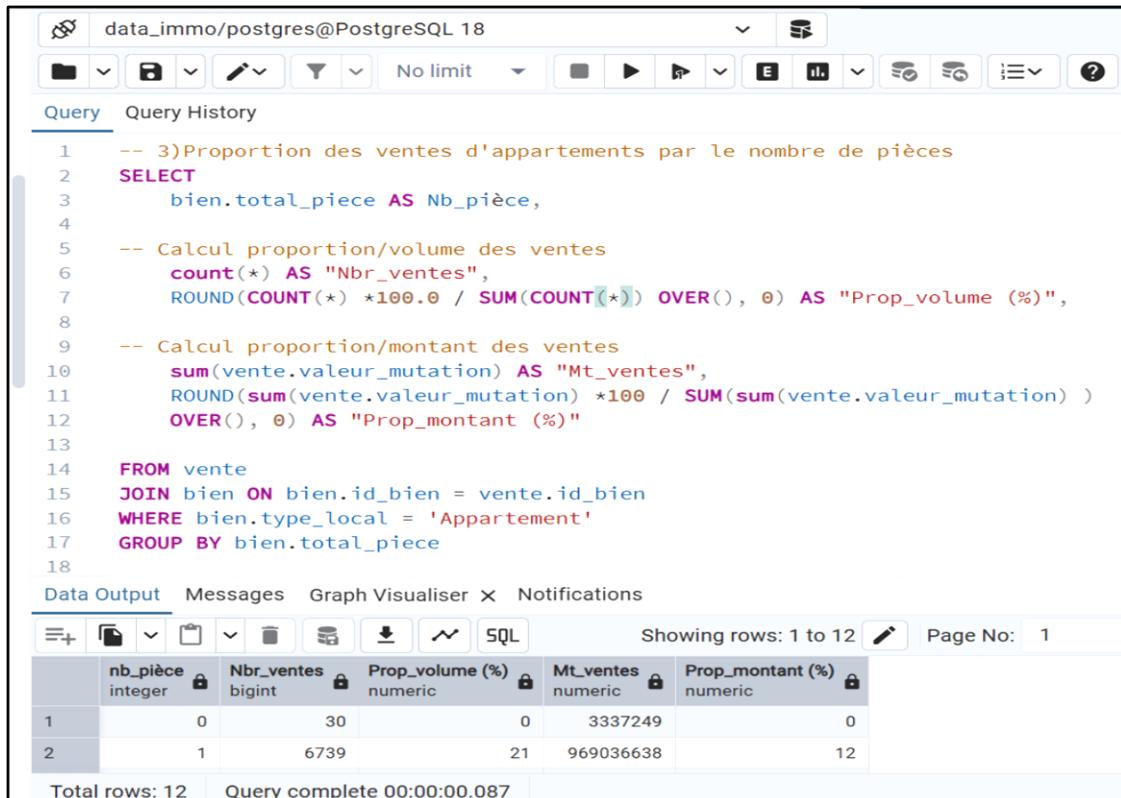
-- 2)Le nombre de ventes d'appartement pour le 1er semestre 2020 par région

```
SELECT
    region.reg_code AS «
        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
```

Résultat

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

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



The screenshot shows a PostgreSQL database interface with the following details:

- Connection:** data_immo/postgres@PostgreSQL 18
- Toolbar:** Includes icons for file operations, search, and various database functions.
- Query History:** Shows the history of previous queries.
- Current Query:**

```

1 -- 3)Proportion des ventes d'appartements par le nombre de pièces
2 SELECT
3     bien.total_piece AS Nb_pièce,
4
5     -- Calcul proportion/volume des ventes
6     count(*) AS "Nbr_ventes",
7     ROUND(COUNT(*) *100.0 / SUM(COUNT(*)) OVER(), 0) AS "Prop_volume (%)",
8
9     -- Calcul proportion/montant des ventes
10    sum(vente.valeur_mutation) AS "Mt_ventes",
11    ROUND(sum(vente.valeur_mutation) *100 / SUM(sum(vente.valeur_mutation)) OVER(), 0) AS "Prop_montant (%)"
12
13
14 FROM vente
15 JOIN bien ON bien.id_bien = vente.id_bien
16 WHERE bien.type_local = 'Appartement'
17 GROUP BY bien.total_piece
18

```
- Data Output:** Shows the results of the query in a table format.
- Table Headers:** nb_pièce, Nbr_ventes, Prop_volume (%), Mt_ventes, Prop_montant (%).
- Table Data:**

	nb_pièce	Nbr_ventes	Prop_volume (%)	Mt_ventes	Prop_montant (%)
1	0	30	0	3337249	0
2	1	6739	21	969036638	12
- Bottom Status:** Total rows: 12, Query complete 00:00:00.087.

Script SQL

```
-- 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 vte (%)",
```

```
-- 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 vte (%)"
```

```
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
```

Résultat

Nb_pièces	Nb_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

Analyse des points clés

1. Le "Cœur du Marché" (2 et 3 pièces) :

- Ils représentent environ **60% du volume** des ventes (**31.18 + 28.57**)
- C'est la preuve que le marché des appartements est dominé par les petites et moyennes surfaces.

2. Le basculement Volume vs Montant :

- Les **3 pièces** : ils représentent **28,57%** des ventes, mais **30,89%** de la valeur totale. C'est le moment où la courbe de valeur dépasse celle du volume.
- Pour les **1 pièce**, c'est l'inverse : beaucoup de ventes (**21,48%**) mais un poids financier plus faible (**12,34%**).

3. L'anomalie des "0 pièces" :

- On a 30 ventes avec "0 pièce". En immobilier (données DVF), cela correspond souvent à des chambres de service non déclarées, des lofts en plateau brut, ou des erreurs de saisie. C'est très bien de les avoir laissés, cela montre l'intégrité des données.

4. La rareté du luxe :

- À partir de 6 pièces, le volume devient infime (moins de 1%), mais le montant reste significatif.

5. Un point de vigilance (Le 10 et 11 pièces)

- Pour 10 pièces : 2 ventes pour un montant total de 657 000€. Cela donne une moyenne de 328 500€ par appartement.
- Le prix de 139 000 € pour un appartement de 11 pièces, est économiquement incohérent pour un bien de cette envergure.
- **Interprétation** : Ces lignes correspondent à des valeurs aberrantes (outliers). Il peut s'agir d'erreurs de saisie dans la base DVF, de ventes de biens en état de ruine, ou de transactions particulières (ventes à prix symbolique, rachats de parts ou viagers).
- **Impact** : Bien que ces données soient présentes dans la base brute, leur très faible volume (1 à 2 ventes) n'altère pas la fiabilité statistique globale des autres catégories (1 à 5 pièces) qui représentent plus de 98% du marché."

Quelques détails techniques :

- **Double SUM** : Pour le pourcentage en valeur, on utilise **SUM(SUM(valeur_mutation)) OVER()** : Cela signifie : "*Fais la somme de toutes les sommes par groupe*".
- **Tri** : Le ORDER BY bien.total_piece permet de lire les résultats logiquement, du plus petit appartement au plus grand.

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

The screenshot shows a PostgreSQL database interface with the following details:

- Query History:** Shows the SQL query being run.
- Data Output:** Displays the results of the query in a table format.
- Table Headers:** dep_nom, surface, valeurs, Prix m².
- Table Data:**

	dep_nom	surface	valeurs	Prix m ²
1	Paris	248447	2973890342	11969.92
2	Hauts-de-Seine	101100	744006157	7359.11
3	Val-de-Marne	106448	536164441	5036.87
4	Alpes-Maritimes	71892	331274231	4607.94
5	Seine-Saint-Denis	55995	230731268	4120.57
6	Hauts-de-Savoie	20176	81635975	4046.16
Total rows: 10				
Query complete 00:00:00.208				

Script

```
-- 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",
    -- On ajoute ::NUMERIC pour permettre l'arrondi
    ROUND((sum(vente.valeur_mutation) / sum(bien.surface_carrez)) , 2) AS "Prix m2"

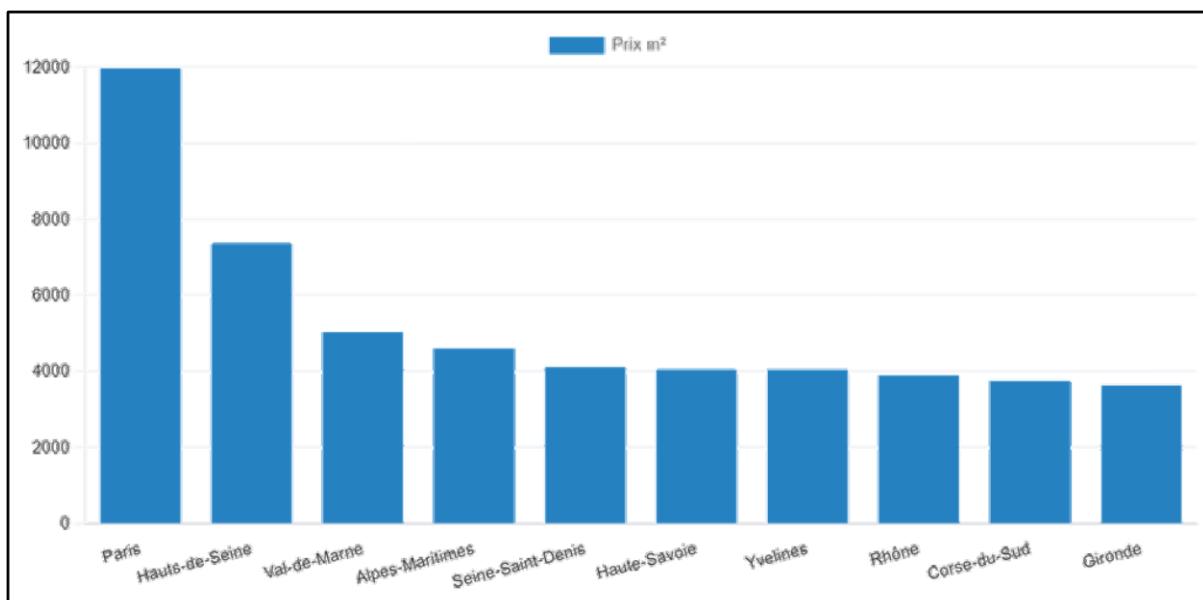
FROM vente
JOIN bien ON bien.id_bien = vente.id_bien
JOIN commune ON commune.code_dep_com = bien.code_dep_com
JOIN departement ON departement.dep_code = commune.dep_code
-- Évite la division par zéro
WHERE bien.surface_carrez > 0
GROUP BY departement.dep_code, departement.dep_nom
ORDER BY "Prix m2" DESC LIMIT 10;
```

Résultat

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

Graphique :

Titre : Concentration des valeurs immobilières (S1/2020) : Hégémonie parisienne



On voit très clairement la hiérarchie du marché immobilier français :

- **Paris** explose les compteurs (aux alentours de 12 000 €/m²).
- Le prix au m² à Paris est 1,6 fois plus élevé que dans les Hauts-de-Seine (2ème) et 3 fois plus élevé qu'en Gironde (10ème).
- Le prix du m² a t'il été impacté par le Covid19?

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

The screenshot shows a PostgreSQL client interface with the following details:

- Query History:** Shows the SQL query being run.
- SQL Editor:** Contains the following SQL code:

```
--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_local),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 ;
```
- Data Output:** Displays the results of the query in a table:

Région	Prix moyen m ²
Île-de-France	3997.71

Script

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

```
SELECT
    region.reg_nom AS "Région",
    -- ROUND(sum(vente.valeur_mutation)/ sum(bien.surface_local),2) AS "Prix moyen m²"
    ROUND(AVG(vente.valeur_mutation/bien.surface_local ),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 ;
```

Résultat

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

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

The screenshot shows a PostgreSQL client interface with the following details:

- Connection:** data_immo/postgres@PostgreSQL 18
- Toolbar:** Includes icons for file operations, search, and various database functions.
- Query History:** A tab labeled "Query History" is selected.
- SQL Query:**

```

1 SELECT
2     region.reg_nom AS "Région",
3     departement.dep_code AS "Code département",
4     departement.dep_nom AS "Nom département",
5     bien.surface_local AS "Surface",
6     vente.valeur_mutation AS "Mt vente",
7     ROUND(vente.valeur_mutation / bien.surface_local, 0) AS "Prix au m²"
8 FROM vente
9 JOIN bien ON bien.id_bien = vente.id_bien
10 JOIN commune ON commune.code_dep_com = bien.code_dep_com
11 JOIN departement ON departement.dep_code = commune.dep_code
12 JOIN region ON region.reg_code = departement.reg_code
13 WHERE bien.type_local = 'Appartement'
14 ORDER BY vente.valeur_mutation DESC
15 LIMIT 10;
    
```
- Data Output:** A table showing the results of the query. The columns are:

	Région character varying (50)	Code département character varying (3)	Nom département character varying (50)	Surface numeric (10,2)	Mt vente numeric (15,2)	Prix au m² numeric
1	Ile-de-France	75	Paris	10.00	9000000.00	900000
2	Ile-de-France	91	Essonne	62.00	8600000.00	138710
3	Ile-de-France	75	Paris	289.00	8577713.00	29681
4	Ile-de-France	75	Paris	42.00	7620000.00	181429
5	Ile de France	75	Paris	200.00	7600000.00	38000
- Total rows: 10**
- Query complete 00:00:00.118**
- CRLF**

Script

--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'
    
```

```
ORDER BY vente.valeur_mutation DESC  

LIMIT 10;
```

Résultat

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

☞ Les valeurs aberrantes en rouge

La réalité du marché de l'ultra-luxe

- Argument** : "À cette époque, les prix plafonnaient rarement au-dessus de **35 000 € à 40 000 €/m²** pour des appartements, sauf cas exceptionnels de terrasses ou vues monumentales."
- Justification** : En mettant le curseur à **50 000**, on reste "large" : on garde tout le luxe réel, mais on élimine les erreurs de saisie manifestes (comme l'appartement à 900 000 €/m²).

Ajustement du code avec une condition

--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_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 au 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'
```

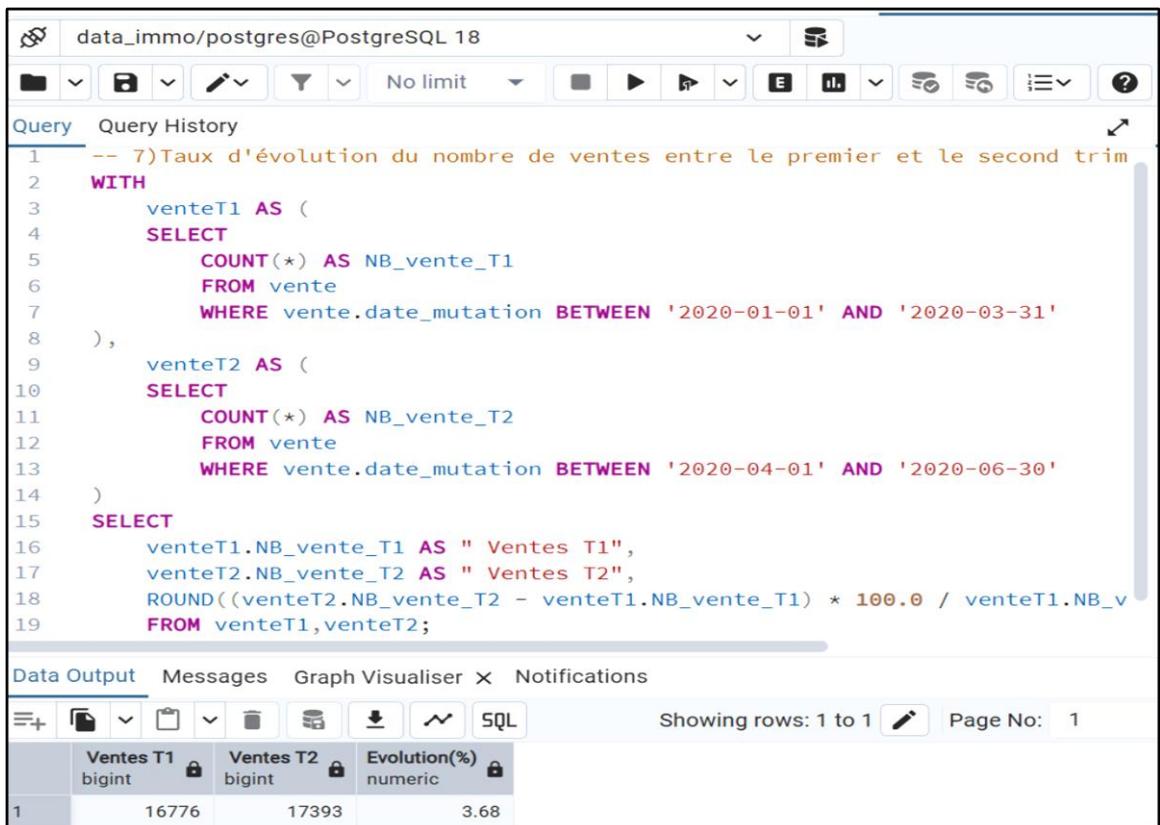
```
-- On exclut les montants aberrants
AND (vente.valeur_mutation / bien.surface_carrez < 40000
ORDER BY vente.valeur_mutation DESC
LIMIT 10;
```

Résultat

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

☞ Données plus cohérentes

7. 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 trim
1 WITH
2     venteT1 AS (
3         SELECT
4             COUNT(*) AS NB_vente_T1
5             FROM vente
6             WHERE vente.date_mutation BETWEEN '2020-01-01' AND '2020-03-31'
7         ),
8     venteT2 AS (
9         SELECT
10            COUNT(*) AS NB_vente_T2
11            FROM vente
12            WHERE vente.date_mutation BETWEEN '2020-04-01' AND '2020-06-30'
13        )
14     SELECT
15         venteT1.NB_vente_T1 AS " Ventes T1",
16         venteT2.NB_vente_T2 AS " Ventes T2",
17         ROUND((venteT2.NB_vente_T2 - venteT1.NB_vente_T1) * 100.0 / venteT1.NB_vente_T1) AS Evolution(%)
18     FROM venteT1,venteT2;
```

	Ventes T1	Ventes T2	Evolution(%)
1	16776	17393	3.68

Script

```
-- 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'
    )

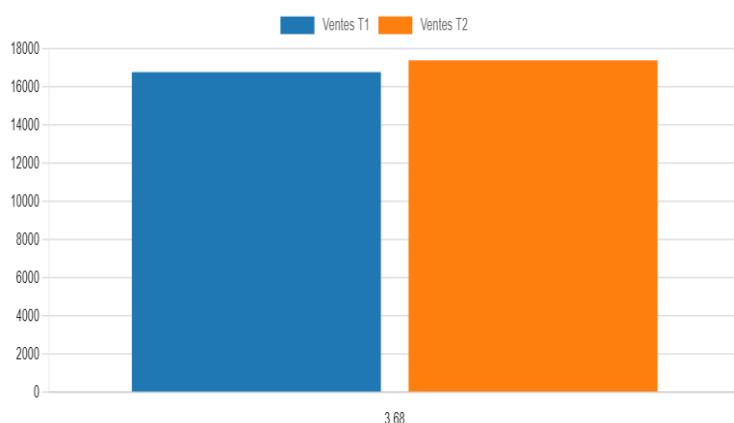
SELECT
    venteT1.NB_vente_T1 AS "Ventes T1",
    venteT2.NB_vente_T2 AS "Ventes T2",
    ROUND((venteT2.NB_vente_T2 - venteT1.NB_vente_T1) * 100.0 /
    venteT1.NB_vente_T1, 2) AS "Evolution(%)"

FROM venteT1,venteT2;
```

Résultat

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

Graphique



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

The screenshot shows the pgAdmin 4 interface with a query editor and a data output viewer.

Query Editor:

```
-- 8)Le classement des régions par rapport au prix au mètre carré des appartements de plus de 4 pièces.
SELECT
    region.reg_nom AS "Nom région",
    COUNT(vente.id_vente) AS "NB vents",
    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 COUNT(vente.id_vente) DESC;
```

Data Output:

	Nom région character varying (50)	NB vents bigint	prix m²(F5 et +) numeric
1	Ile-de-France	685	9171.26
2	Auvergne-Rhône-Alpes	185	2932.00
3	Provence-Alpes-Côte d'A...	117	3166.99
4	Grand Est	76	1475.14
5	Nouvelle-Aquitaine	68	2545.21

Total rows: 15 Query complete 00:00:00.123 CRLF Ln 5, Col 70

Script

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

SELECT
    reg_nom AS "Région",
    COUNT(vente.id_vente) AS "NB vents",
    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 "NB ventes" DESC;
```

Résultat

Nom région	NB vents	prix m ² (F5 et +)
Ile-de-France	685	9171,26
Auvergne-Rhône-Alpes	185	2932
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

1) Analyse

a. La confirmation statistique

- **L'Ile-de-France** (685 ventes) : Le chiffre de 8 997 €/m² est extrêmement solide. Avec un tel volume, c'est une donnée de marché très fiable.
- **Auvergne-Rhône-Alpes** (185 ventes) et **PACA** (117 ventes) : Ces régions ont également des volumes suffisants pour que leurs moyennes soient considérées comme représentatives.

b. Les "Alertes" sur la fiabilité (Échantillons faibles)

- **La Réunion** (2 ventes) et la **Corse** (5 ventes) : Bien qu'elles soient 2ème et 3ème, leurs prix sont basés sur un nombre infime de transactions. Une seule vente de luxe a pu faire bondir la moyenne.
- **La Martinique** (1 seule vente) : Comme nous l'avions pressenti, le prix de 564 €/m² n'est pas "le prix du marché", c'est juste le prix d'un seul appartement vendu cette année-là. Ce n'est pas statistiquement représentatif de la région.

2) Ajustement de la requête pour un meilleur résultat (bon échantillon)

Dans un rapport d'analyse, on ajoute souvent une clause HAVING pour exclure les régions où le volume est trop faible, afin de ne pas fausser les conclusions.

Par exemple, si on voulait ne garder que les régions avec un "marché significatif" (plus de 10 ventes), on pourrait ajouter un clause ainsi

GROUP BY region.reg_nom

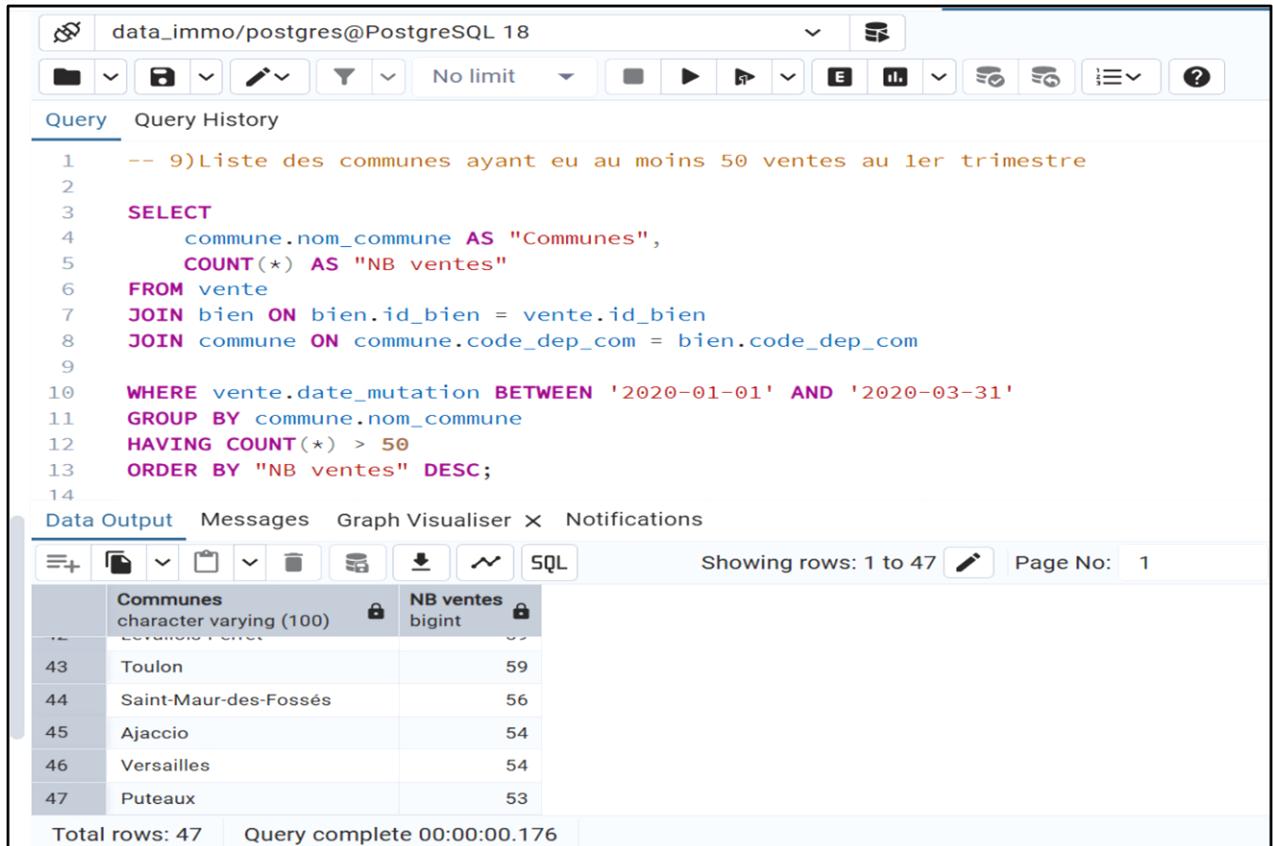
HAVING COUNT(vente.id_vente) > 10 -- Exclut les régions avec trop peu de données

ORDER BY "prix m²" DESC;

Nouveau résultat

Nb	Région	NB ventes	prix m ²
1	Ile-de-France	685	8997,17
2	Auvergne-Rhône-Alpes	185	2863,18
3	Provence-Alpes-Côte d'Azur	117	3072,43
4	Grand Est	76	1409,73
5	Nouvelle-Aquitaine	68	2607,18
6	Pays de la Loire	54	2400,36
7	Occitanie	47	2214,27
8	Hauts-de-France	40	2296,88
9	Bretagne	37	2304,61
10	Centre-Val de Loire	31	1451,92
11	Bourgogne-Franche-Comté	26	1063,85
12	Normandie	26	2108

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



The screenshot shows a PostgreSQL database interface with the following details:

- Connection:** data_immo/postgres@PostgreSQL 18
- Toolbar:** Includes icons for file operations, search, filters, and various database management functions.
- Query Bar:** Shows the query history and the current query being run.
- Query Content:**

```
1 -- 9)Liste des communes ayant eu au moins 50 ventes au 1er trimestre
2
3 SELECT
4     commune.nom_commun AS "Communes",
5     COUNT(*) AS "NB ventes"
6 FROM vente
7 JOIN bien ON bien.id_bien = vente.id_bien
8 JOIN commune ON commune.code_dep_com = bien.code_dep_com
9
10 WHERE vente.date_mutation BETWEEN '2020-01-01' AND '2020-03-31'
11 GROUP BY commune.nom_commun
12 HAVING COUNT(*) > 50
13 ORDER BY "NB ventes" DESC;
14
```
- Data Output:** A table showing the results of the query:

	Communes character varying (100)	NB ventes bigint
43	Toulon	59
44	Saint-Maur-des-Fossés	56
45	Ajaccio	54
46	Versailles	54
47	Puteaux	53

Total rows: 47 | Query complete 00:00:00.176

Script

```
-- 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;
```

Résultat

Communes	NB ventes
Paris 17e Arrondissement	228
Paris 15e Arrondissement	215
Paris 18e Arrondissement	209
Nice	173
Paris 11e Arrondissement	169
Paris 16e Arrondissement	165
Bordeaux	157
Paris 14e Arrondissement	146
Paris 20e Arrondissement	127
Nantes	119
Paris 19e Arrondissement	116
Paris 12e Arrondissement	110
Paris 10e Arrondissement	109
Paris 9e Arrondissement	106
Grenoble	106
Boulogne-Billancourt	99
Paris 13e Arrondissement	94
Paris 7e Arrondissement	87
Paris 6e Arrondissement	86
Asnières-sur-Seine	81
Marseille 8e Arrondissement	81

Courbevoie	80
Paris 3e Arrondissement	79
Paris 5e Arrondissement	79
Toulouse	78
Antibes	77
Marseille 4e Arrondissement	72
Marseille 1er Arrondissement	71
Vincennes	68
Rueil-Malmaison	68
Lille	67
Marseille 9e Arrondissement	66
Montreuil	65
Angers	64
Nîmes	63
La Ciotat	62
Paris 8e Arrondissement	62
Sète	62
Paris 2e Arrondissement	61
Rennes	61
Paris 4e Arrondissement	60
Levallois-Perret	59
Toulon	59
Saint-Maur-des-Fossés	56
Ajaccio	54
Versailles	54
Puteaux	53
"Issy-les-Moulineaux"	50

Analyse des résultats

- **La domination parisienne** : Paris écrase le classement. Le fait que vous ayez des résultats par arrondissement (17e, 15e, 18e en tête) prouve que votre jointure avec la table commune fonctionne parfaitement, car Paris est géré par codes Insee d'arrondissements dans les données DVF.
- **Les métropoles régionales** : On retrouve logiquement les grandes locomotives comme Nice, Bordeaux, Nantes et Grenoble.
- **Le dynamisme de la petite couronne** : La présence de Boulogne-Billancourt, Asnières-sur-Seine et Courbevoie montre le report du marché parisien sur les communes limitrophes.
- **Le Sud** : Antibes, Marseille (par arrondissement également), Nîmes et Ajaccio confirment que le littoral reste une zone de fort volume de transactions, même en hiver.

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

The screenshot shows a PostgreSQL client interface with the following details:

- Connection:** data_immo/postgres@PostgreSQL 18
- Query History:**

```

5  SELECT round(sum(vente.valeur_mutation)/sum(bien.surface_local),2) AS "% Prix moyen Appart T2"
6  FROM vente
7  JOIN bien ON bien.id_bien = vente.id_bien
8  WHERE bien.type_local = 'Appartement'
9  GROUP BY bien.total_piece HAVING bien.total_piece = 2
10 ),
11
12 PM_appart_T3 AS (
13  SELECT round(sum(vente.valeur_mutation)/sum(bien.surface_local),2) AS "% Prix moyen Appart T3"
14  FROM vente
15  JOIN bien ON bien.id_bien = vente.id_bien
16  WHERE bien.type_local = 'Appartement'
17  GROUP BY bien.total_piece HAVING bien.total_piece = 3
18 )
19
20 SELECT "% Prix moyen Appart T2", "% Prix moyen Appart T3",
21       round(("%" Prix moyen Appart T2" -%" Prix moyen Appart T3") * 100.0 /
22             ("%" Prix moyen Appart T3"),2) AS "Ecart %"
23 FROM PM_appart_T2, PM_appart_T3;

```
- Data Output:**

	% Prix moyen Appart T2 numeric	% Prix moyen Appart T3 numeric	Ecart % numeric
1	4763.41	4228.27	12.66
- Message Bar:** Total rows: 1 Query complete 00:00:00.077 CRLF Ln 20, Col 34

Script 1

-- 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;

```

Résultat

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

Analyse du résultat : 14,31%

- Le résultat de 14,31% % confirme une règle d'or de l'immobilier : plus la surface d'un bien est petite, plus son prix au mètre carré est élevé.
- Les T2 sont extrêmement demandés par les investisseurs locatifs et les jeunes actifs, ce qui tire les prix vers le haut par rapport aux T3, plus orientés vers les familles.

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

The screenshot shows the pgAdmin 4 interface with a query editor and a results grid.

Query Editor:

```
data_immo/postgres@PostgreSQL 18
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;
```

Data Output:

	Département character varying (3)	Commune character varying (100)	Prix moyen numeric
1	06	Saint-Jean-Cap-Ferrat	968750.0
2	06	Eze	655000.0

Total rows: 15 Query complete 00:00:00.314 CRLF

Script SQL

```
-- 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;
```

Aperçu

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
----	------------------------	------------

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

The screenshot shows a pgAdmin 4 interface with a query editor and a results grid. The query is:

```

1 -- 12)Les 20 communes avec le plus de transactions immobilières
2
3 SELECT
4     nom_commune,
5     COUNT(*) AS "nombre de transactions"
6 FROM vente
7 JOIN bien ON bien.id_bien = vente.id_bien
8 JOIN commune ON commune.code_dep_com = bien.code_dep_com
9 GROUP BY commune.nom_commune
10 ORDER BY "nombre de transactions"
11 DESC
12 LIMIT 20;
  
```

The results grid displays the top 20 communes with the highest number of transactions, ordered by descending count. The columns are 'nom_commune' and 'nombre de transactions'.

	nom_commune	nombre de transactions
1	Paris 18e Arrondissement	516
2	Paris 17e Arrondissement	470
3	Paris 15e Arrondissement	407
4	Paris 16e Arrondissement	394
5	Nice	393
6	Paris 11e Arrondissement	383
	Total rows: 20	Query complete 00:00:00.422

Script

```

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

    code_postal,
    nom_commune as "nom commune",
    pop_totale AS "Population",
    round(avg(surface_carrez),0) as "Surface Carrez moyenne",
    round(avg(valeur_mutation),0) as "Valeur foncière moyenne",
    round(round((count(id_bien))*1000,2)/pop_totale,2) as "Nombre d'achat pour
mille habitants"
FROM vente
JOIN bien USING (id_bien)
JOIN commune USING (code_dep_com)
WHERE surface_carrez != 0
AND dep_code IS NOT NULL AND pop_totale > 10000
GROUP BY nom_commune, code_postal, pop_totale
ORDER BY "Nombre d'achat pour mille habitants" DESC
  
```

LIMIT 20;

Aperçu

Code postal	Nom commune	Population	Surface Carrez moyenne	Valeur foncière moyenne	Nombre d'achat pour mille habitants
75002	Paris 2e Arrondissement	21 735,00	42	521065	5.84
75001	Paris 1er Arrondissement	16 055,00	54	1000675	4.92
75003	Paris 3e Arrondissement	34 306,00	49	655954	4.69
33120	Arcachon	11 898,00	50	307436	4.62
44500	La Baule-Escoublac	16 797,00	50	264745	4.58
75004	Paris 4e Arrondissement	29 390,00	48	645485	4.08
06190	Roquebrune-Cap-Martin	13 041,00	47	274173	3.99
75008	Paris 8e Arrondissement	36 250,00	75	1010536	3.83
83110	Sanary-sur-Mer	17 160,00	48	268489	3.50
75009	Paris 9e Arrondissement	60 563,00	50	583177	3.43
83250	La Londe-les-Maures	10 776,00	37	151213	3.43
75006	Paris 6e Arrondissement	41 171,00	59	1005956	3.38
83270	Saint-Cyr-sur-Mer	11 725,00	53	256894	3.24
60500	Chantilly	11 178,00	61	238481	3.13
44380	Pornichet	11 440,00	44	204127	3.06
94160	Saint-Mandé	22 576,00	55	494061	3.06
75010	Paris 10e Arrondissement	86 863,00	49	532282	3.04
06500	Menton	30 981,00	45	208585	2.94
85270	Saint-Hilaire-de-Riez	11 501,00	36	85265	2.87
94300	Vincennes	50 230,00	50	433108	2.81

Code optimisé

```
-- 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;

```

Résultat

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