

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

```
1 • SELECT COUNT(vente.Id_bien) as nombre_appart_vendu
2 FROM vente
3 JOIN bien ON vente.Id_bien = bien.Id_bien
4 WHERE bien.Type_local = 'Appartement' AND DATE(Date_mutation) BETWEEN '2020-01-01' AND '2020-06-30';
5
```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:	Result Grid
nombre_appart_vendu				Form Editor
31362				Field Types

Result 1 x Read Only

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

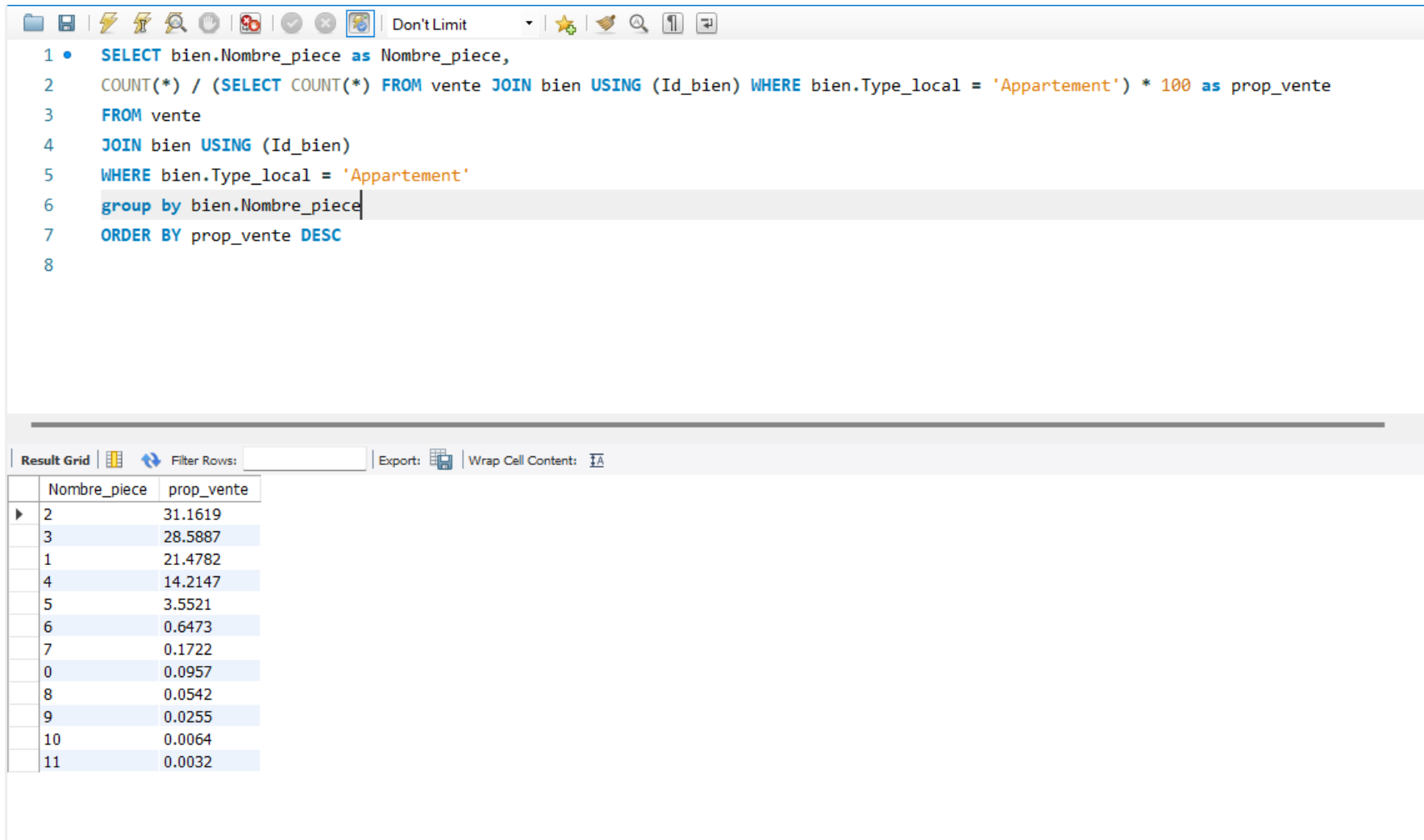
```
1 • SELECT reg_nom as Nom_region, COUNT(vente.Id_bien) as Nombre_vente
2 FROM vente
3 JOIN bien USING (Id_bien)
4 JOIN commune USING (id_codedep_codecommune)
5 JOIN region USING (reg_code)
6 WHERE bien.Type_local = 'Appartement' AND DATE(Date_mutation) BETWEEN '2020-01-01' AND '2020-06-30'
7 group by Nom_region
8 ORDER BY Nombre_vente DESC
```

Result Grid | Filter Rows: | Export: | Wrap Cell Contents: |

Nom_region	Nombre_vente
Ile-de-France	13989
Provence-Alpes-Côte d'Azur	3645
Auvergne-Rhône-Alpes	3253
Nouvelle-Aquitaine	1931
Occitanie	1640
Pays de la Loire	1357
Hauts-de-France	1252
Grand Est	984
Bretagne	983
Normandie	861
Centre-Val de Loire	695
Bourgogne-Franche-Comté	376
Corse	222
Martinique	94
La Réunion	44
Guyane	34
Guadeloupe	2

Result 3 x

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



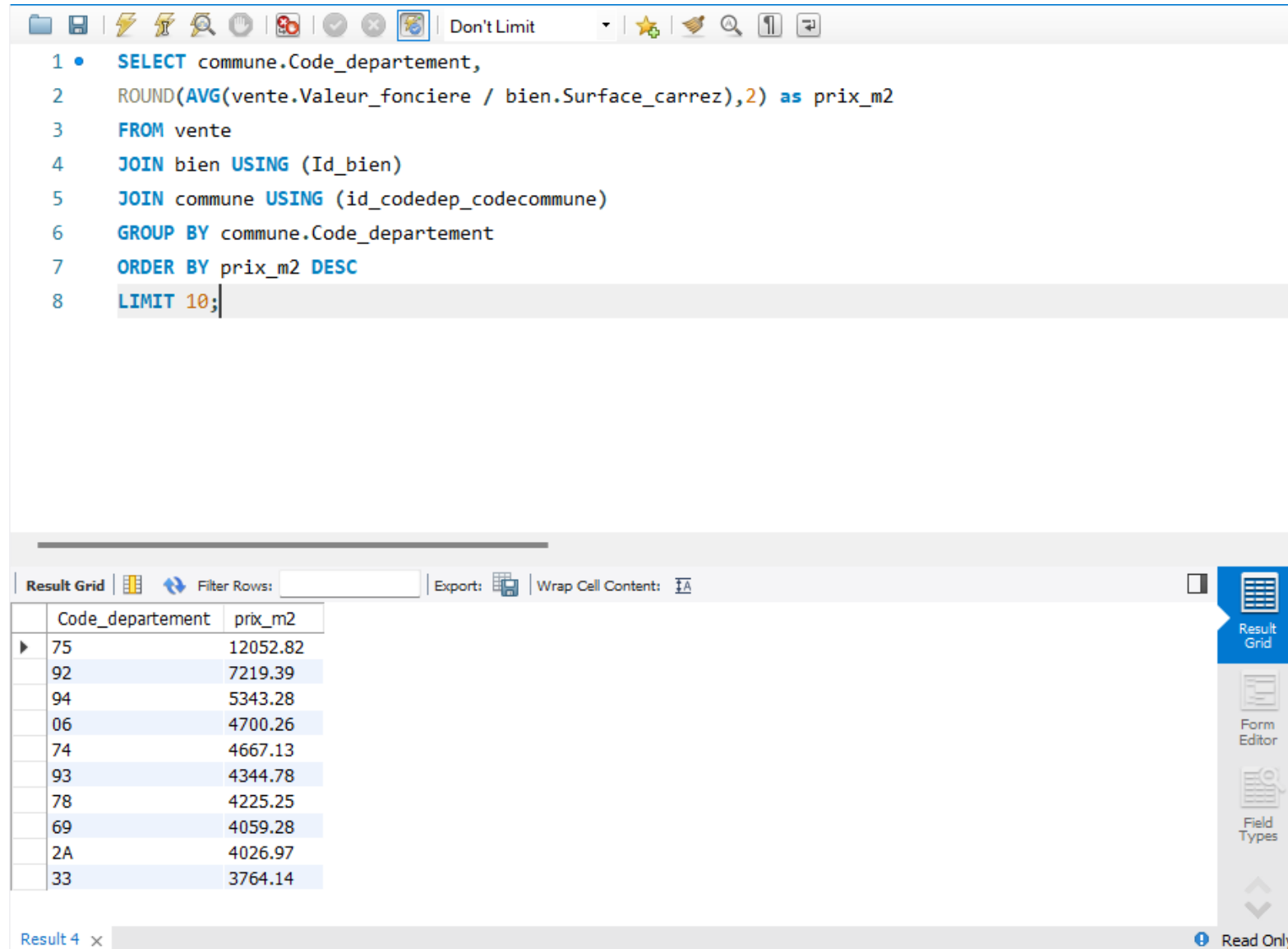
The screenshot shows a SQL query editor window with a toolbar at the top. The query is as follows:

```
1 • SELECT bien.Nombre_piece as Nombre_piece,  
2   COUNT(*) / (SELECT COUNT(*) FROM vente JOIN bien USING (Id_bien) WHERE bien.Type_local = 'Appartement') * 100 as prop_vente  
3 FROM vente  
4 JOIN bien USING (Id_bien)  
5 WHERE bien.Type_local = 'Appartement'  
6 group by bien.Nombre_piece  
7 ORDER BY prop_vente DESC  
8
```

Below the query editor is a results grid. The grid has two columns: 'Nombre\_piece' and 'prop\_vente'. The data is sorted in descending order of 'prop\_vente'.

	Nombre_piece	prop_vente
▶	2	31.1619
	3	28.5887
	1	21.4782
	4	14.2147
	5	3.5521
	6	0.6473
	7	0.1722
	0	0.0957
	8	0.0542
	9	0.0255
	10	0.0064
	11	0.0032

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



The screenshot shows a database query editor interface. The top toolbar includes icons for file operations, execution, and settings. The query text is as follows:

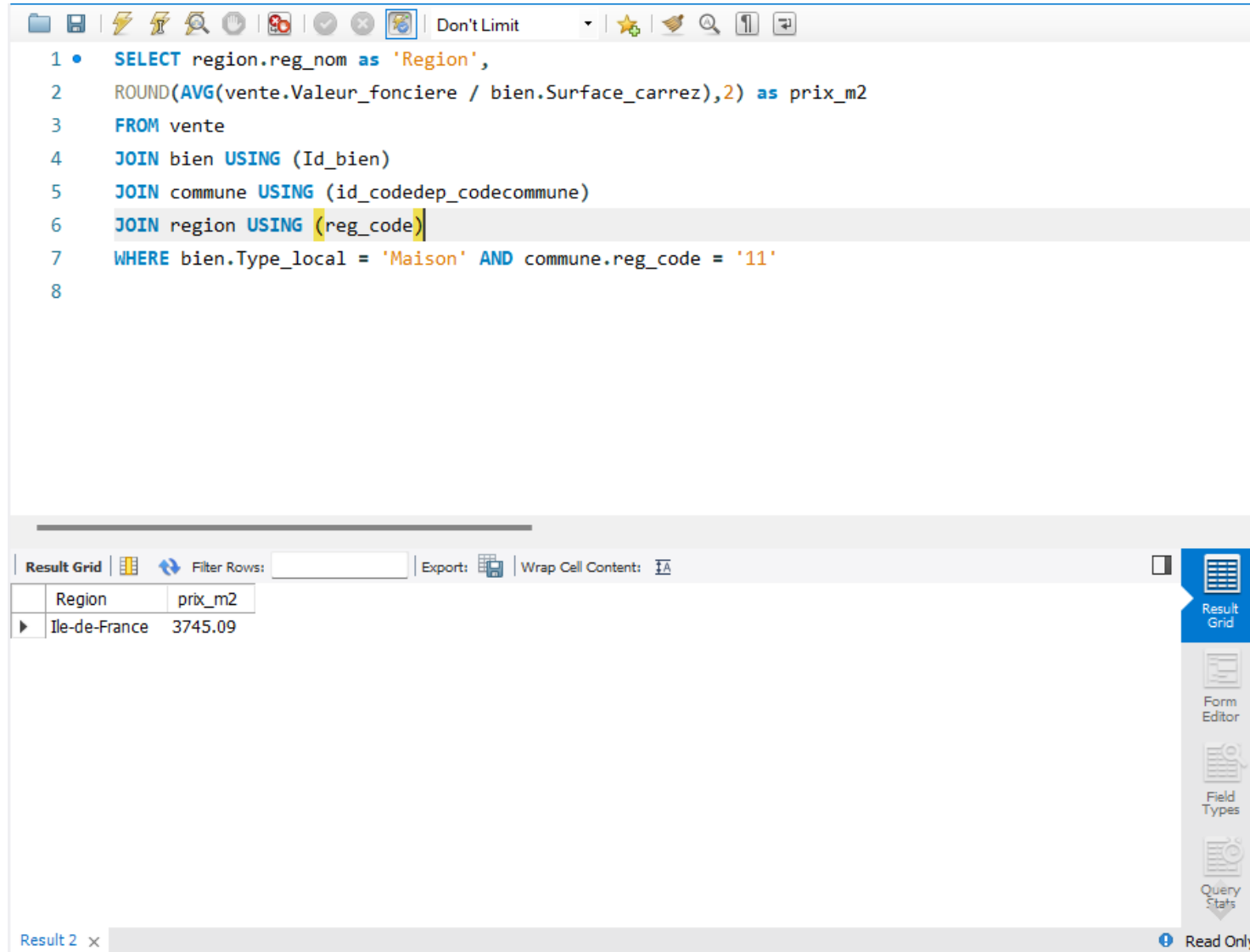
```
1 • SELECT commune.Code_departement,  
2   ROUND(AVG(vente.Valeur_fonciere / bien.Surface_carrez),2) as prix_m2  
3 FROM vente  
4 JOIN bien USING (Id_bien)  
5 JOIN commune USING (id_codedep_codecommune)  
6 GROUP BY commune.Code_departement  
7 ORDER BY prix_m2 DESC  
8 LIMIT 10;
```

Below the query editor, the 'Result Grid' tab is active, displaying the results of the query. The results are shown in a table with two columns: 'Code\_departement' and 'prix\_m2'. The table contains 10 rows of data, sorted by 'prix\_m2' in descending order.

Code_departement	prix_m2
75	12052.82
92	7219.39
94	5343.28
06	4700.26
74	4667.13
93	4344.78
78	4225.25
69	4059.28
2A	4026.97
33	3764.14

The bottom of the interface shows a tab labeled 'Result 4' and a 'Read Only' status indicator.

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



The screenshot shows a database query editor interface. The top toolbar includes icons for file operations, execution, and search. The SQL query is as follows:

```
1 • SELECT region.reg_nom as 'Region',  
2 ROUND(AVG(vente.Valeur_fonciere / bien.Surface_carrez),2) as prix_m2  
3 FROM vente  
4 JOIN bien USING (Id_bien)  
5 JOIN commune USING (id_codedep_codecommune)  
6 JOIN region USING (reg_code)  
7 WHERE bien.Type_local = 'Maison' AND commune.reg_code = '11'  
8
```

Below the query editor, the 'Result Grid' tab is active, displaying the following data:

Region	prix_m2
▶ Ile-de-France	3745.09

The right sidebar contains icons for 'Result Grid', 'Form Editor', 'Field Types', and 'Query Stats'. The bottom status bar indicates 'Result 2' and 'Read Only'.

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

```
1 SELECT bien.Id_bien as 'Bien', bien.Surface_carrez as Surface, vente.Valeur_fonciere as 'Valeur Foncière', region.reg_nom as 'Region'
2 FROM vente
3 JOIN bien USING (Id_bien)
4 JOIN commune USING (id_codedep_codecommune)
5 JOIN region USING (reg_code )
6 WHERE bien.Type_local = 'Appartement'
7 ORDER BY vente.Valeur_fonciere DESC
8 LIMIT 10;
```

	Bien	Surface	Valeur Foncière	Region
▶	30576	9.1	9000000	Ile-de-France
	5258	64	8600000	Ile-de-France
	3623	20.55	8577713	Ile-de-France
	7598	42.77	7620000	Ile-de-France
	9983	253.3	7600000	Ile-de-France
	17814	139.9	7535000	Ile-de-France
	410	360.95	7420000	Ile-de-France
	16348	595	7200000	Ile-de-France
	1923	122.56	7050000	Ile-de-France
	19150	79.38	6600000	Ile-de-France

Result 5 ×

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

The screenshot shows a database query editor with a SQL query and its results. The query calculates the percentage increase in sales from the first quarter to the second quarter of 2020.

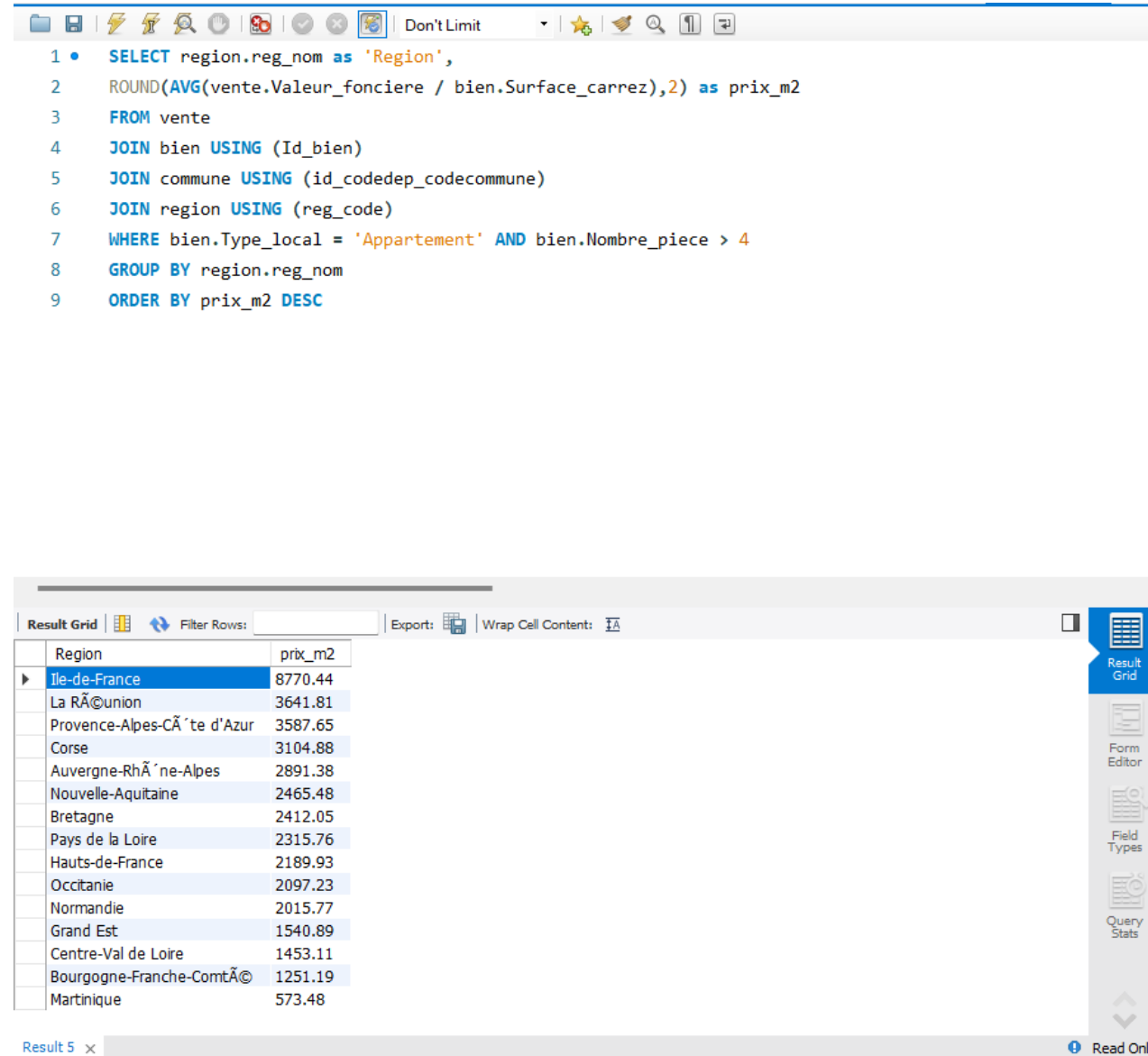
```
1 • SELECT (COUNT(vente.Id_vente) -  
2     (SELECT COUNT(vente.Id_vente) as premier_trim  
3     FROM vente  
4     WHERE DATE(vente.Date_mutation) BETWEEN '2020-01-01' AND '2020-03-31')) /  
5     (SELECT COUNT(vente.Id_vente) as premier_trim  
6     FROM vente  
7     WHERE DATE(vente.Date_mutation) BETWEEN '2020-01-01' AND '2020-03-31')  
8     as Taux_Evolution  
9 FROM vente  
10 WHERE DATE(vente.Date_mutation) BETWEEN '2020-04-01' AND '2020-06-30'
```

The results are displayed in a table with one row:

Taux_Evolution
0.0366

The interface includes a toolbar at the top with icons for file operations, a status bar at the bottom indicating 'Result 3' and 'Read Only', and a sidebar on the right with icons for 'Result Grid', 'Form Editor', 'Field Types', and 'Query Stats'.

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



The screenshot shows a SQL query editor window with a toolbar at the top. The query is as follows:

```
1 • SELECT region.reg_nom as 'Region',  
2   ROUND(AVG(vente.Valeur_fonciere / bien.Surface_carrez),2) as prix_m2  
3 FROM vente  
4 JOIN bien USING (Id_bien)  
5 JOIN commune USING (id_codedep_codecommune)  
6 JOIN region USING (reg_code)  
7 WHERE bien.Type_local = 'Appartement' AND bien.Nombre_piece > 4  
8 GROUP BY region.reg_nom  
9 ORDER BY prix_m2 DESC
```

Below the query editor, the 'Result Grid' tab is active, displaying the results of the query. The table has two columns: 'Region' and 'prix\_m2'. The results are sorted in descending order of 'prix\_m2'.

Region	prix_m2
Ile-de-France	8770.44
La R��union	3641.81
Provence-Alpes-C��te d'Azur	3587.65
Corse	3104.88
Auvergne-Rh��ne-Alpes	2891.38
Nouvelle-Aquitaine	2465.48
Bretagne	2412.05
Pays de la Loire	2315.76
Hauts-de-France	2189.93
Occitanie	2097.23
Normandie	2015.77
Grand Est	1540.89
Centre-Val de Loire	1453.11
Bourgogne-Franche-Comt��	1251.19
Martinique	573.48

At the bottom of the window, there is a status bar showing 'Result 5' and a 'Read Only' indicator.



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

Nom_Commune	nombre_vente
AJACCIO	54
ANGERS	64
ANTIBES	77
ASNIERES-SUR-SEINE	81
BORDEAUX	157
BOULOGNE-BILLANCOURT	99
COURBEVOIE	80
GRENOBLE	106
ISSY-LES-MOULINEAUX	50
LA CIOTAT	62
LEVALLOIS-PERRET	59
LILLE	67
MARSEILLE 1ER ARRONDI...	71
MARSEILLE 4E ARRONDI...	72
MARSEILLE 8E ARRONDI...	81
MARSEILLE 9E ARRONDI...	66
MONTREUIL	65
NANTES	119
NICE	173
NIMES	63
PARIS 10E ARRONDISSE...	109
PARIS 11E ARRONDISSE...	169
PARIS 12E ARRONDISSE...	110
PARIS 13E ARRONDISSE...	94
PARIS 14E ARRONDISSE...	146
PARIS 15E ARRONDISSE...	215
PARIS 16E ARRONDISSE...	165
PARIS 17E ARRONDISSE...	228
PARIS 18E ARRONDISSE...	209
PARIS 19E ARRONDISSE...	116
PARIS 20E ARRONDISSE...	127
PARIS 2E ARRONDISSEM...	61
PARIS 3E ARRONDISSEM...	79
PARIS 4E ARRONDISSEM...	59
PARIS 5E ARRONDISSEM...	79

```

1 SELECT commune.Nom_commune as Nom_Commune, COUNT(vente.Id_vente) as nombre_vente
2 FROM vente
3 JOIN bien USING (Id_bien)
4 JOIN commune USING (id_codedep_codecommune)
5 WHERE DATE(vente.Date_mutation) BETWEEN '2020-01-01' AND '2020-04-01'
6 GROUP BY commune.Nom_commune
7 HAVING nombre_vente >= 50
8 ORDER BY Nom_Commune asc

```

PARIS 3E ARRONDISSEM...	79
PARIS 4E ARRONDISSEM...	59
PARIS 5E ARRONDISSEM...	79
PARIS 6E ARRONDISSEM...	86
PARIS 7E ARRONDISSEM...	87
PARIS 8E ARRONDISSEM...	62
PARIS 9E ARRONDISSEM...	106
PUTEAUX	53
RENNES	61
RUEIL-MALMAISON	68
SAINT-MAUR-DES-FOSSES	56
SETE	62
TOULON	59
TOULOUSE	78
VERSAILLES	54
VINCENNES	68

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

Différence en % d'un 3 pièces à un 2 pièces :

```
1 • SELECT (ROUND(AVG(vente.Valeur_fonciere / bien.Surface_carrez),2) /
2 (SELECT ROUND(AVG(vente.Valeur_fonciere / bien.Surface_carrez),2)
3 FROM vente
4 JOIN bien USING (Id_bien)
5 WHERE bien.Nombre_piece = 3 AND bien.Type_local = 'Appartement') * 100) - 100 as diff_prix_m2
6 FROM vente
7 JOIN bien USING (Id_bien)
8 WHERE bien.Nombre_piece = 2 AND bien.Type_local = 'Appartement'
```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
diff_prix_m2			
▶ 14.155976445854293			

Différence en % d'un 2 pièces à un 3 pièces :

```
1 • SELECT (ROUND(AVG(vente.Valeur_fonciere / bien.Surface_carrez),2) /
2 (SELECT ROUND(AVG(vente.Valeur_fonciere / bien.Surface_carrez),2)
3 FROM vente
4 JOIN bien USING (Id_bien)
5 WHERE bien.Nombre_piece = 2 AND bien.Type_local = 'Appartement') * 100) - 100 as diff_prix_m2
6 FROM vente
7 JOIN bien USING (Id_bien)
8 WHERE bien.Nombre_piece = 3 AND bien.Type_local = 'Appartement'
```

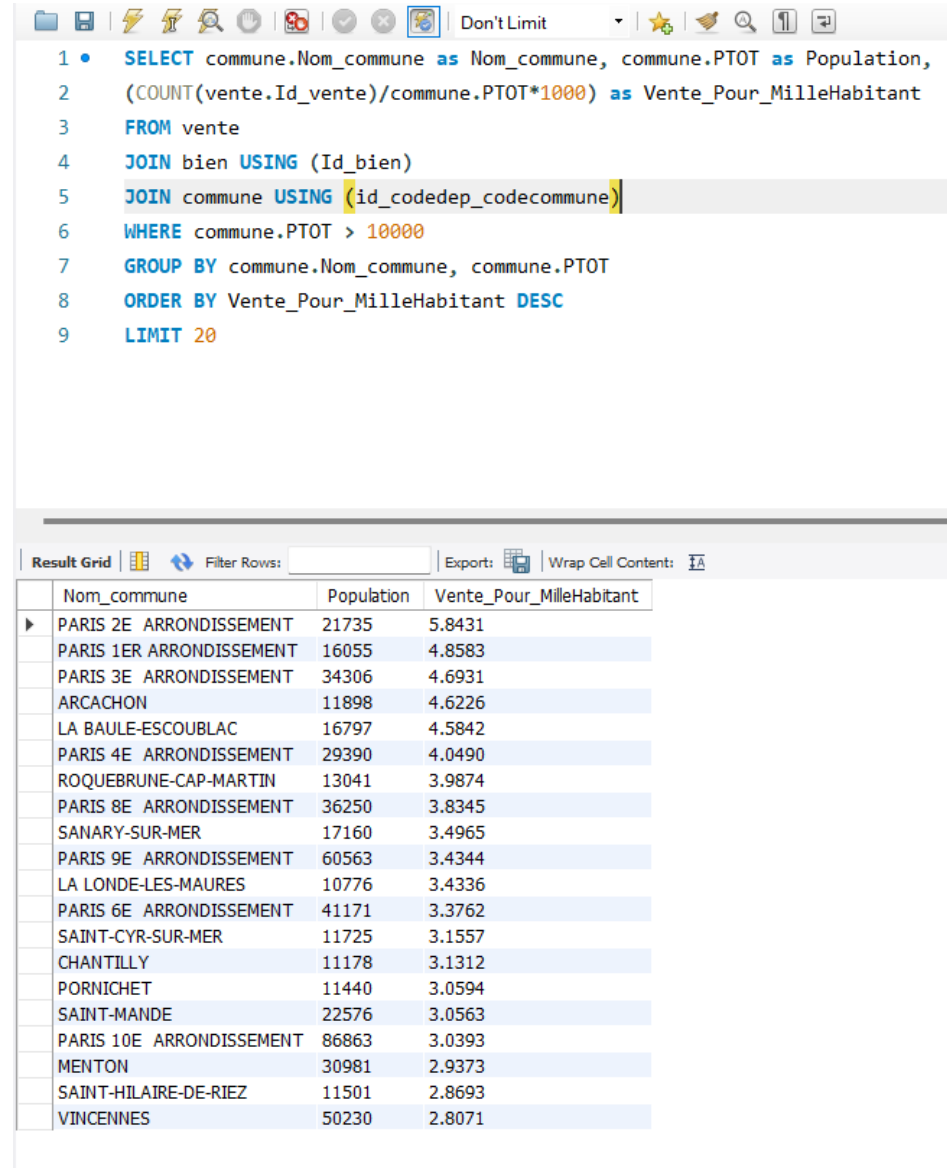
Result Grid	Filter Rows:	Export:	Wrap Cell Content:
diff_prix_m2			
▶ -12.40055657757575			

# Requête n°11 : Les moyennes de valeurs foncières pour le top 3 des communes des départements 6, 13, 33, 59 et 69

```
1 • WITH Tri as (  
2     SELECT commune.Nom_commune as Nom_Commune,  
3     ROUND(AVG(vente.Valeur_fonciere),2) as Valeur_Fonciere,  
4     commune.Code_departement as Code_Departement,  
5     ROW_NUMBER() OVER (PARTITION BY commune.Code_departement ORDER BY AVG(vente.Valeur_fonciere) DESC) as Ordre  
6     FROM vente  
7     JOIN bien USING (Id_bien)  
8     JOIN commune USING (id_codedep_codecommune)  
9     WHERE Code_Departement IN ('06', '13', '33', '59', '69')  
10    GROUP BY Nom_Commune, Code_Departement)  
11  
12    SELECT  
13        Tri.Nom_Commune as 'Nom Commune',  
14        Tri.Valeur_Fonciere as 'Moyenne Valeur Foncière',  
15        Tri.Code_Departement as 'Code département',  
16        Ordre  
17    FROM Tri  
18    WHERE Ordre <= 3  
19    ORDER BY Tri.Code_Departement
```

	Nom Commune	Moyenne Valeur Foncière	Code département	Ordre
►	GIGNAC-LA-NERTHE	330000.00	13	1
	SAINT-SAVOURNIN	314425.00	13	2
	CASSIS	313416.88	13	3
	LEGE-CAP-FERRET	549500.64	33	1
	VAYRES	335000.00	33	2
	ARCACHON	307435.93	33	3
	BERSEE	433202.00	59	1
	CYSOING	408550.00	59	2
	HALLUIN	322250.00	59	3
	VILLE-SUR-JARNIOUX	485300.00	69	1
	LYON 2E ARRONDISSEMENT	455217.26	69	2
	LYON 6E ARRONDISSEMENT	426968.25	69	3

# Requête n°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 SQL query editor window with a toolbar at the top. The query is as follows:

```
1 • SELECT commune.Nom_commune as Nom_commune, commune.PTOT as Population,
2 (COUNT(vente.Id_vente)/commune.PTOT*1000) as Vente_Pour_MilleHabitant
3 FROM vente
4 JOIN bien USING (Id_bien)
5 JOIN commune USING (id_codedep_codecommune)
6 WHERE commune.PTOT > 10000
7 GROUP BY commune.Nom_commune, commune.PTOT
8 ORDER BY Vente_Pour_MilleHabitant DESC
9 LIMIT 20
```

Below the query editor is a 'Result Grid' window. It has a toolbar with 'Filter Rows', 'Export', and 'Wrap Cell Content' options. The results are displayed in a table with the following columns: Nom\_commune, Population, and Vente\_Pour\_MilleHabitant. The table contains 20 rows of data, sorted by the 'Vente\_Pour\_MilleHabitant' column in descending order.

Nom_commune	Population	Vente_Pour_MilleHabitant
PARIS 2E ARRONDISSEMENT	21735	5.8431
PARIS 1ER ARRONDISSEMENT	16055	4.8583
PARIS 3E ARRONDISSEMENT	34306	4.6931
ARCACHON	11898	4.6226
LA BAULE-ESCOUBLAC	16797	4.5842
PARIS 4E ARRONDISSEMENT	29390	4.0490
ROQUEBRUNE-CAP-MARTIN	13041	3.9874
PARIS 8E ARRONDISSEMENT	36250	3.8345
SANARY-SUR-MER	17160	3.4965
PARIS 9E ARRONDISSEMENT	60563	3.4344
LA LONDE-LES-MAURES	10776	3.4336
PARIS 6E ARRONDISSEMENT	41171	3.3762
SAINT-CYR-SUR-MER	11725	3.1557
CHANTILLY	11178	3.1312
PORNICHET	11440	3.0594
SAINT-MANDE	22576	3.0563
PARIS 10E ARRONDISSEMENT	86863	3.0393
MENTON	30981	2.9373
SAINT-HILAIRE-DE-RIEZ	11501	2.8693
VINCENNES	50230	2.8071