

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

Requête :

```
SELECT COUNT(valeur) AS Nombre_ventes_1er_semestre
FROM vente
JOIN bien ON bien.id_bien = vente.id_bien
WHERE typ_loc = 'Appartement' AND mut_dat BETWEEN '01-01-2020 00:00:00' AND
'30-06-2020 00:00:00';
```

Résultat :

	nombre_ventes_1er_semestre bigint
1	31364
Total rows: 1 of 1    Query complete 00:00:00.087	

## 2. Le nombre de ventes d'appartement par région pour le 1<sup>er</sup> semestre 2020.

Requête :

```
SELECT region.reg_nom AS Région, COUNT(valeur) AS Nombre_de_ventes
FROM vente
JOIN bien ON bien.id_bien = vente.id_vente
JOIN commune ON commune.id_coddep_codcom = bien.id_coddep_codcom
JOIN region ON region.cod_dep = commune.cod_dep
WHERE bien.typ_loc = 'Appartement'
GROUP BY region.reg_nom
ORDER BY Nombre_de_ventes DESC;
```

Résultat :

	<div>région</div> <div>character varying (50)</div>	<div>nombre_de_ventes</div> <div>bigint</div>
1	Ile-de-France	13985
2	Provence-Alpes-Côte d'Azur	3644
3	Auvergne-Rhône-Alpes	3252
4	Nouvelle-Aquitaine	1930
5	Occitanie	1640
6	Pays de la Loire	1357
7	Hauts-de-France	1252
8	Grand Est	984
9	Bretagne	983
10	Normandie	861
11	Centre-Val de Loire	695
12	Bourgogne-Franche-Comté	376
13	Corse	222
14	Martinique	94
15	La Réunion	44
16	Guyane	34
17	Guadeloupe	2
Total rows: 17 of 17		Query complete 00:00:00.127

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

Requête :

```
SELECT bien.nb_pieces AS nombre_pieces, COUNT(DISTINCT
vente.id_vente)*100/(SELECT COUNT(vente.id_vente) FROM vente) AS proportion
FROM vente
JOIN bien ON bien.id_bien = vente.id_vente
WHERE typ_loc = 'Appartement' AND nb_pieces !=0
GROUP BY bien.nb_pieces;
```

Résultat :

	nombre_pieces integer	proportion bigint
1	1	19
2	2	28
3	3	26
4	4	13
5	5	3
6	6	0
7	7	0
8	8	0
9	9	0
10	10	0
11	11	0

Total rows: 11 of 11    Query complete 00:00:00.125

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

Requête :

```
SELECT region.dep_nom AS département, ROUND (SUM(valeur)/SUM(surf_rel),2) AS
prix_mètre_carré
FROM region
JOIN commune ON region.cod_dep = commune.cod_dep
JOIN bien ON commune.id_coddep_codcom = bien.id_coddep_codcom
JOIN vente ON bien.id_bien = vente.id_vente
GROUP BY dep_nom
ORDER BY prix_mètre_carré DESC
LIMIT 10;
```

Résultat :

	département character varying (50)	prix_mètre_carré numeric
1	Paris	11950.99
2	Hauts-de-Seine	7332.08
3	Val-de-Marne	5040.02
4	Alpes-Maritimes	4613.96
5	Seine-Saint-Denis	4162.07
6	Haute-Savoie	4051.33
7	Yvelines	4039.56
8	Rhône	3903.09
9	Guadeloupe	3894.74
10	Corse-du-Sud	3729.38

Total rows: 10 of 10    Query complete 00:00:00.141

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

Requête :

```
SELECT 'Maison' AS type_logement, ROUND(AVG(valeur/surf_car),2) AS Prix_moyen
FROM vente
JOIN bien ON bien.id_bien = vente.id_vente
JOIN commune ON commune.id_coddep_codcom = bien.id_coddep_codcom
JOIN region ON region.cod_dep = commune.cod_dep
WHERE bien.typ_loc='Maison' AND reg_nom ='Ile-de-France';
```

Résultat :

	type_logement text	prix_moyen numeric
1	Maison	2485.71
Total rows: 1 of 1		Query complete 00:00:00.064

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

Requête :

```
SELECT bien.id_bien, reg_nom AS région, valeur AS prix, surf_rel AS
nombre_mètres_carrés
FROM region
JOIN commune ON region.cod_dep = commune.cod_dep
JOIN bien ON commune.id_coddep_codcom = bien.id_coddep_codcom
JOIN vente ON bien.id_bien = vente.id_vente
WHERE valeur IS not null AND typ_loc='Appartement'
ORDER BY valeur DESC
LIMIT 10;
```

Résultat :

	id_bien integer	région character varying (50)	prix numeric	nombre_mètres_carrés integer
1	32275	Ile-de-France	9000000	46
2	21835	Ile-de-France	8600000	72
3	29799	Ile-de-France	8577710	108
4	32433	Ile-de-France	7620000	27
5	29850	Ile-de-France	7600000	20
6	29522	Ile-de-France	7535000	18
7	31973	Ile-de-France	7420000	170
8	32135	Ile-de-France	7200000	34
9	29353	Ile-de-France	7050000	61
10	29513	Ile-de-France	6600000	45
Total rows: 10 of 10		Query complete 00:00:00.125		

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

Requête :

```
WITH VENTE1 AS (  
    SELECT COUNT(vente.valeur) AS vente1  
    FROM vente  
    JOIN bien ON bien.id_bien = vente.id_bien  
    WHERE typ_loc = 'Appartement' AND mut_dat BETWEEN '2020-01-01' AND  
'2020-03-31 00:00:00')  
, VENTE2 AS(  
    SELECT COUNT(valeur) AS vente2  
    FROM vente  
    JOIN bien ON bien.id_bien = vente.id_bien  
    WHERE typ_loc = 'Appartement' AND mut_dat BETWEEN '2020-04-01' AND  
'2020-06-30 00:00:00')  
SELECT vente1 AS Ventes_1er_tri, vente2 AS Ventes_2eme_tri,  
    ROUND((CAST(vente2 AS Numeric)-CAST(vente1 AS  
Numeric))*100/CAST(vente1 AS Numeric),2) AS Taux_évolution  
FROM VENTE1  
CROSS JOIN VENTE2;
```

Résultat :

	ventes_1er_tri bigint	ventes_2eme_tri bigint	taux_évolution numeric
1	15387	15977	3.83
Total rows: 1 of 1    Query complete 00:00:00.084			

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

Requête :

```
SELECT reg_nom AS Région, ROUND(AVG(valeur/ surf_car),2) AS prix_mètre_carré  
FROM region  
JOIN commune ON region.cod_reg = commune.cod_reg  
JOIN bien ON commune.id_coddep_codcom = bien.id_coddep_codcom  
JOIN vente ON bien.id_bien = vente.id_bien  
WHERE surf_car != 0  
    AND typ_loc = 'Appartement'  
    AND nb_pieces > 4  
GROUP BY reg_nom  
ORDER BY prix_mètre_carré DESC;
```

Résultat :

	<b>région</b> character varying (50)	<b>prix_mètre_carré</b> numeric
1	Ile-de-France	8619.12
2	La Réunion	3641.81
3	Corse	3104.88
4	Provence-Alpes-Côte d'Azur	3087.20
5	Auvergne-Rhône-Alpes	2600.80
6	Nouvelle-Aquitaine	2465.48
7	Bretagne	2412.05
Total rows: 15 of 15    Query complete 00:00:00.072		

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

Requête :

```
WITH ventes_tri1 AS (  
  SELECT commune.com, COUNT(vente.valeur) AS nb_ventes  
  FROM commune  
    JOIN bien on commune.id_coddep_codcom = bien.id_coddep_codcom  
    JOIN vente on vente.id_bien = bien.id_bien  
 WHERE mut_dat BETWEEN '01-01-2020 00:00:00' AND '31-03-2020 00:00:00'  
  GROUP BY commune.com  
  HAVING COUNT(vente.valeur) >= 50  
)  
SELECT *  
FROM ventes_tri1  
ORDER BY nb_ventes DESC;
```

Résultat :

	<b>commune</b> character varying (50)	<b>nombre_ventes</b> bigint
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
Total rows: 48 of 48    Query complete 00:00:00.096		

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

Requête :

```
WITH prix_m2_2p AS (  
    SELECT ROUND((SUM(vente.valeur) / SUM(bien.surf_car)), 2) AS prix_m2_2p  
    FROM bien  
    JOIN vente ON bien.id_bien = vente.id_bien  
    WHERE typ_loc = 'Appartement' AND nb_pieces = 2),  
prix_m2_3p AS (  
    SELECT ROUND((SUM(vente.valeur) / SUM(bien.surf_car)), 2) AS prix_m2_3p  
    FROM bien  
    JOIN vente ON bien.id_bien = vente.id_bien  
    WHERE typ_loc = 'Appartement' AND nb_pieces = 3)  
SELECT prix_m2_2p, prix_m2_3p, ABS(ROUND(((prix_m2_3p - prix_m2_2p) * 100 /  
prix_m2_2p), 2)) AS difference_pourcentage  
FROM prix_m2_2p  
CROSS JOIN prix_m2_3p;
```

Résultat :

	prix_m2_2p numeric	prix_m2_3p numeric	difference_pourcentage numeric
1	4809.76	4112.33	14.50
Total rows: 1 of 1    Query complete 00:00:00.120			

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

Requête :

```
WITH valeur_par_ville AS (  
  SELECT cod_dep, commune.com, AVG(valeur) AS valeur  
  FROM vente  
  JOIN bien USING (id_bien)  
  JOIN commune USING (id_coddep_codcom)  
  WHERE commune.cod_dep IN ('6', '13', '33', '59', '69')  
  GROUP BY cod_dep, commune.com)  
SELECT code_departement AS "Département", nom_commune AS "Commune",  
  ROUND(CAST(valeur AS Numeric),2) AS "Prix moyen"  
FROM (  
  SELECT cod_dep AS code_departement, com AS nom_commune, valeur,  
    rank() OVER (PARTITION BY cod_dep ORDER BY valeur DESC) AS rang  
  FROM valeur_par_ville) AS result  
WHERE rang <= 3
```

Résultat :

	Département character varying (10)	Commune character varying (50)	Prix moyen numeric
1	13	Gignac-la-Nerthe	330000.00
2	13	Saint-Savournin	314425.00
3	13	Cassis	313416.88
4	33	Lège-Cap-Ferret	549500.64
5	33	Vayres	335000.00
6	33	Arcachon	307435.93
7	59	Bersée	433202.00
8	59	Cysoing	408550.00
9	59	Halluin	322250.00
10	6	Saint-Jean-Cap-Ferrat	968750.00
11	6	Eze	655000.00
12	6	Mouans-Sartoux	476898.09
13	69	Saint-Germain-au-Mont-d'Or	490000.00
14	69	Ville-sur-Jarnioux	485300.00
15	69	Lyon 6e Arrondissement	426968.25
Total rows: 15 of 15    Query complete 00:00:00.075			

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

Requête :

```
WITH commune_transact AS (  
  
    SELECT commune.com AS commune, COUNT(vente.valeur) AS nb_transactions,  
    commune.ptot  
  
    FROM commune  
  
    JOIN bien ON commune.id_coddep_codcom = bien.id_coddep_codcom  
  
    JOIN vente ON bien.id_bien = vente.id_vente  
  
    WHERE commune.ptot > 10000  
  
    GROUP BY commune.com, commune.ptot  
  
)  
  
SELECT commune, ROUND((nb_transactions * 1000.0) / ptot, 2) AS  
transactions_per_1000  
  
FROM commune_transact  
  
ORDER BY transactions_per_1000 DESC  
  
LIMIT 20;
```

Résultat :

	commune character varying (50) 🔒	transactions_per_1000 numeric 🔒
1	Paris 2e Arrondissement	5.75
2	Paris 1er Arrondissement	4.92
3	Paris 3e Arrondissement	4.69
4	Arcachon	4.62
5	La Baule-Escoublac	4.58
6	Paris 4e Arrondissement	4.08
7	Roquebrune-Cap-Martin	3.99
8	Paris 8e Arrondissement	3.81
9	Sanary-sur-Mer	3.5
10	La Londe-les-Maures	3.43
11	Paris 9e Arrondissement	3.43
12	Paris 6e Arrondissement	3.38
13	Saint-Cyr-sur-Mer	3.24
14	Chantilly	3.13
15	Pornichet	3.06
16	Paris 10e Arrondissement	3.04
17	Saint-Mandé	3.01
18	Menton	2.94
19	Saint-Hilaire-de-Riez	2.87
20	Vincennes	2.81