

- - - - - REQUETES SQL - - - - -

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

```
SELECT COUNT ( id_vente ) AS ventes_appartements_1er_Semetre_2020
FROM ventes JOIN biens ON biens.id_bien = ventes.id_bien
WHERE type_local ILIKE 'appartement'
AND date_mutation BETWEEN '2020/01/01' AND '2020/06/30'
```


2 / Nombre de ventes d'appartement, par région,
pour le 1er semestre 2020.

```
SELECT nom_region, COUNT( id_vente ) AS nombre_de_ventes
FROM ventes JOIN biens ON biens.id_bien = ventes.id_bien
JOIN communes ON communes.code_departement_code_commune
= biens.code_departement_code_commune
JOIN regions ON regions.code_region = communes.code_region
WHERE type_local ILIKE 'appartement'
AND date_mutation BETWEEN '2020/01/01' AND '2020/06/30'
GROUP BY nom_region ORDER BY nombre_de_ventes DESC
```


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

```
WITH Total AS ( SELECT COUNT ( id_vente ) AS Total_des_ventes
FROM ventes JOIN biens ON biens.id_bien = ventes.id_bien
WHERE type_local ILIKE 'appartement' )
SELECT nombre_piece,
ROUND ( COUNT ( id_vente ) * 100.00 / ( SELECT Total_des_ventes FROM
Total ), 2 )
AS Pourcentage_des_ventes
FROM ventes JOIN biens ON biens.id_bien = ventes.id_bien
WHERE type_local ILIKE 'appartement'
GROUP BY nombre_piece ORDER BY nombre_piece
```


4 / Liste des dix départements dont le prix du mètre carré est le plus
élevé.

```
SELECT nom_departement,
ROUND ( AVG ( valeur_fonciere / surface_carrez ) ) AS Prix_moyen_m2
FROM ventes JOIN biens ON biens.id_bien = ventes.id_bien
JOIN communes ON communes.code_departement_code_commune
= biens.code_departement_code_commune
JOIN departements ON departements.code_departement
```

```
= communes.code_departement
GROUP BY nom_departement
ORDER BY Prix_moyen_m2 DESC LIMIT 10
```

```
*****
*****
```

5 / Prix moyen du mètre carré pour une maison en Île-de-France.

```
SELECT ROUND( AVG ( valeur_fonciere / surface_carrez ) ) AS
prix_moyen_m2_maison_IDF
FROM ventes JOIN biens ON biens.id_bien = ventes.id_bien
JOIN Communes ON communes.code_departement_code_commune
= biens.code_departement_code_commune
JOIN Regions ON regions.code_region = communes.code_region
WHERE type_local ILIKE 'maison' AND nom_region ILIKE 'ile-de-France'
```

```
*****
*****
```

6 / Liste des dix appartements les plus chers, avec la région et la superficie.

```
SELECT Ventes.id_bien, valeur_fonciere, surface_carrez, nom_region
FROM Ventes JOIN biens ON biens.id_bien = ventes.id_bien
JOIN communes ON communes.code_departement_code_commune
= biens.code_departement_code_commune
JOIN regions ON regions.code_region = communes.code_region
WHERE valeur_fonciere IS NOT NULL
ORDER BY valeur_fonciere DESC LIMIT 10
```

```
*****
*****
```

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

```
WITH Trimestre_1 AS (
    SELECT COUNT( id_vente ) AS trimestre_1 FROM ventes
    WHERE date_mutation BETWEEN '2020/01/01' AND '2020/03/31' ),
    Trimestre_2 AS (
    SELECT COUNT( id_vente ) AS trimestre_2 FROM ventes
    WHERE date_mutation BETWEEN '2020/04/01' AND '2020/06/30' )
SELECT Trimestre_1.trimestre_1 AS ventes_trimestre_1,
    Trimestre_2.trimestre_2 AS ventes_trimestre_2,
    ROUND( ( Trimestre_2.trimestre_2 - Trimestre_1.trimestre_1 ) * 100.00 /
    Trimestre_1.trimestre_1, 2 )
AS Taux_Evolution FROM Trimestre_1, Trimestre_2
```

```
*****
*****
```

8 / Classement des régions par rapport au prix au mètre carré des appartement de plus de 4 pièces.

```

SELECT nom_region,
ROUND( AVG( valeur_fonciere / surface_carrez ) ) AS prix_moyen_m2
FROM ventes JOIN biens ON biens.id_bien = ventes.id_bien
JOIN Communes ON communes.code_departement_code_commune
= biens.code_departement_code_commune
JOIN Regions ON regions.code_region = communes.code_region
WHERE type_local ILIKE 'appartement' AND nombre_piece > 4
GROUP BY nom_region ORDER BY prix_moyen_m2 DESC

```

```

*****
*****

```

9 / Liste des communes enregistrant au moins 50 ventes au 1er trimestre.

```

SELECT nom_commune, COUNT ( ventes.id_vente ) AS nombre_de_ventes
FROM ventes JOIN biens ON biens.id_bien = ventes.id_bien
JOIN communes ON communes.code_departement_code_commune
= biens.code_departement_code_commune
WHERE date_mutation BETWEEN '2020/01/01' AND '2020/03/31'
GROUP BY nom_commune HAVING COUNT( ventes.id_vente ) >= 50
ORDER BY nom_commune ASC

```

```

*****
*****

```

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

```

WITH Appartement_2_pieces AS (
    SELECT ROUND( AVG( valeur_fonciere / surface_carrez ) )
AS prix_moyen_m2_2_pieces
    FROM ventes JOIN biens ON biens.id_bien = ventes.id_bien
    WHERE type_local ILIKE 'appartement' AND nombre_piece = 2 ),
Appartement_3_pieces AS (
    SELECT ROUND( AVG( valeur_fonciere / surface_carrez ) )
AS prix_moyen_m2_3_pieces
    FROM ventes JOIN biens ON biens.id_bien = ventes.id_bien
    WHERE type_local ILIKE 'appartement' AND nombre_piece = 3 )
SELECT Appartement_2_pieces.prix_moyen_m2_2_pieces,
Appartement_3_pieces.prix_moyen_m2_3_pieces,
ROUND( ( Appartement_3_pieces.prix_moyen_m2_3_pieces
- Appartement_2_pieces.prix_moyen_m2_2_pieces )
/ Appartement_2_pieces.prix_moyen_m2_2_pieces * 100.00 )
AS Pourcentage_difference
FROM Appartement_2_pieces, Appartement_3_pieces

```

```

*****
*****

```

11 / Moyennes des valeurs foncières pour le top 3 des communes des départements 6, 13, 33, 59 et 69.

```

SELECT * FROM (
    SELECT nom_departement,
    ROW_NUMBER( ) OVER ( PARTITION BY nom_departement
    ORDER BY ROUND( AVG ( valeur_fonciere ) ) DESC ) AS top3,
    nom_commune, ROUND( AVG( valeur_fonciere ) ) AS
moyenne_valeur_fonciere
    FROM ventes JOIN biens ON biens.id_bien = ventes.id_bien
    JOIN communes ON communes.code_departement_code_commune
    = biens.code_departement_code_commune
    JOIN departements ON departements.code_departement =
communes.code_departement
    WHERE departements.code_departement IN ('6', '13', '33', '59',
'69')
    GROUP BY nom_departement, nom_commune ) AS ranked_data
WHERE top3 <= 3
ORDER BY nom_departement, top3;

```

```

*****
*****

```

12 / Top 20 des communes enregistrant le plus de transactions
pour 1000 habitants, pour les communes de plus de 10 000 habitants.

```

SELECT nom_commune, COUNT( id_vente ) AS nombre_transactions, population,
ROUND( ( COUNT( id_vente ) * 1000.00 / population), 2 ) AS
Transaction_pour_1000_habitants
FROM ventes JOIN biens ON biens.id_bien = ventes.id_bien
JOIN communes ON communes.code_departement_code_commune =
biens.code_departement_code_commune
WHERE population > '10000'
GROUP BY nom_commune, population ORDER BY Transaction_pour_1000_habitants
DESC LIMIT 20

```

```

*****
*****

```