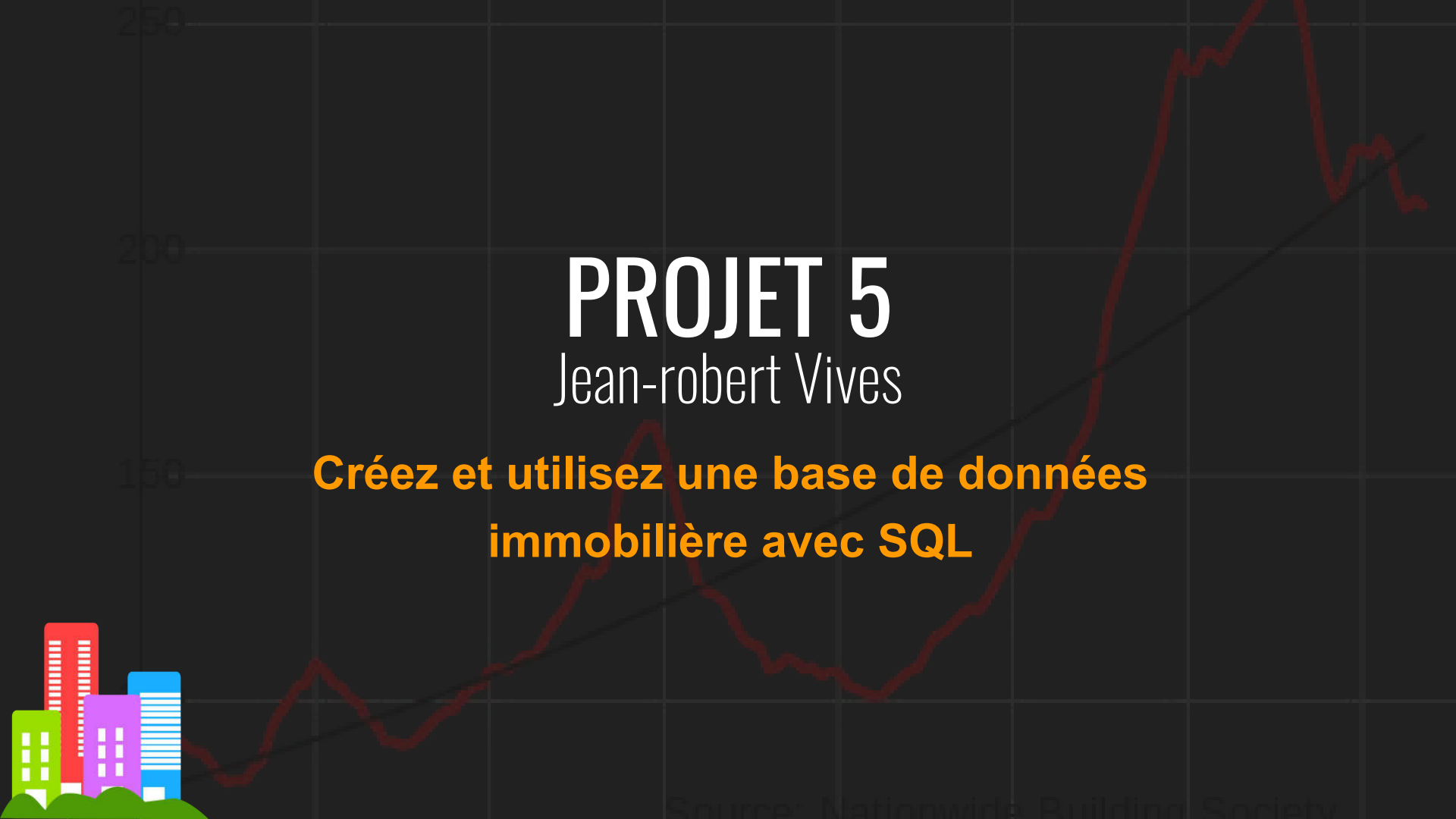




PROJET 5

Jean-robot Vives

**Créez et utilisez une base de données
immobilière avec SQL**

Source: Nationwide Building Society



250
200
150

Analyser le marché de l'agence Laplace Immo
et aider leur agence régionales
dans l'accompagnement de leurs clients.

Source: Nationwide Building Society

1 - Les données à disposition

3 fichiers excel (2020) :

- "Valeurs-foncières" : Les données extraites du site open data des Demandes de valeurs foncières (DVF).
- "donnees_communes" : Les données de l'INSEE avec les résultats des recensements de la population.
- "fr-esr-referentiel-geographique" : Les données data.gouv sur les régions, avec le référentiel géographique français, communes, unités urbaines, aires urbaines, départements, académies, régions.



Nettoyage des données

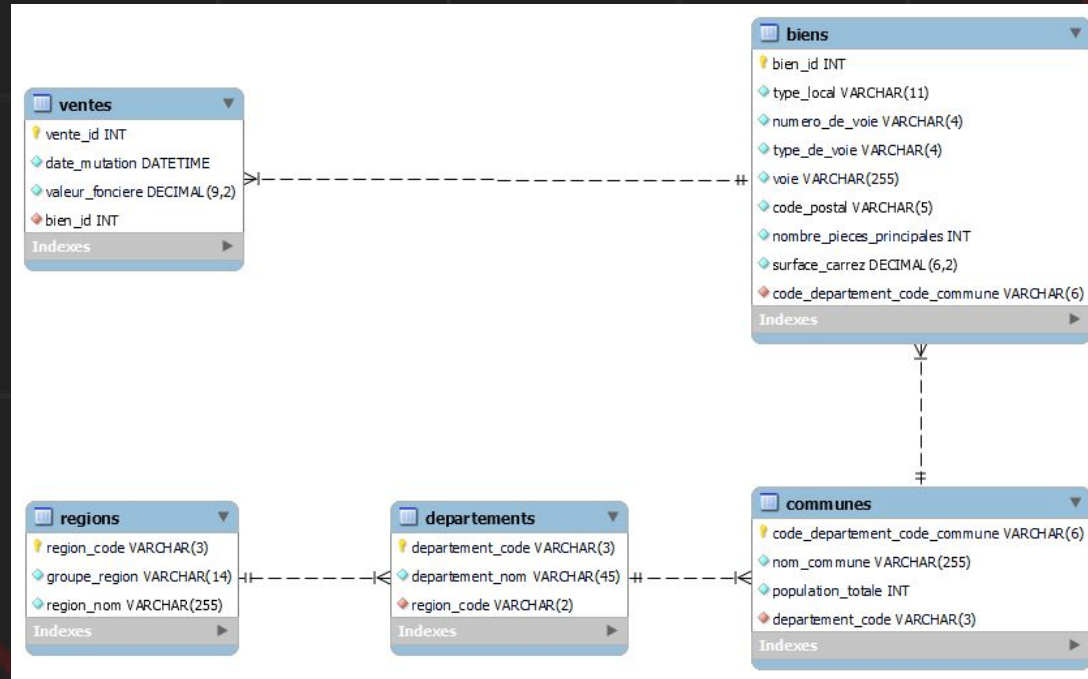
(Nom, valeur nulle, création de nouveaux champs par fusion ...)



2 - Le dictionnaire des données

CODE	SIGNIFICATION	TYPE	LONGUEUR	Clé	REGLE DE GESTION
vente_id	ID de la vente dans la base de données	INT		Primaire	Ne doit pas être nul
date_mutation	Date de la vente	DATE			Ne doit pas être nul
valeur_fonciere	montant du prix du bien vendu	DECIMAL	9.2		Ne doit pas être nul
bien_id	ID du bien dans la base de données	INT		Primaire / Etrangère	Ne doit pas être nul
numero_de_voie	Numéro de la voie du bien	INT	4		Ne doit pas être nul
type_de_voie	Type de voie du bien	VARCHAR	4		Ne doit pas être nul
voie	Nom de la voie du bien	VARCHAR	255		Ne doit pas être nul
code_postal	code postal du bien en vente	VARCHAR	5		Ne doit pas être nul
code_departement_code_commune	ID combiné du code département et du code commune	VARCHAR	6	Primaire / Etrangère	Ne doit pas être nul
surface_carrez	Surface du bien vendu	DECIMAL	6.2		Ne doit pas être nul
type_local	Ce champ indique le type de bien maison ou appartement	VARCHAR	11		Ne doit pas être nul
nombre_pieces_principales	Nombre de pièces principales dans le bien vendu	INT			Ne doit pas être nul
groupe_region	Dénomination DROM-COM/Ile de france / Province	VARCHAR	14		Ne doit pas être nul
region_nom	Nom de la région	VARCHAR	255		Ne doit pas être nul
departement_nom	Nom du département	VARCHAR	45		Ne doit pas être nul
nom_commune	Nom de la commune	VARCHAR	255		Ne doit pas être nul
departement_code	code géographique qui identifie le département	VARCHAR	3	Primaire / Etrangère	Ne doit pas être nul
region_code	Code	VARCHAR	3	Primaire / Etrangère	Ne doit pas être nul
population_totale	Population totale.	INT			Ne doit pas être nul

3 - La création du schéma relationnel



Avec MySQL Workbench

4 - La conformité avec le RGPD



Absence de données personnelles

Lors du traitement des fichiers de travail nous n'avons pas pris en compte "Nom de l'acquéreur" et donc en l'état de notre étude, il est impossible d'attribuer un nom à un bien.

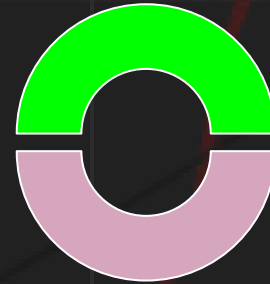
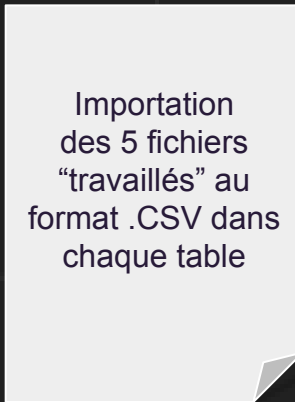
Ensuite les données clients peuvent être gardées [3 ans](#) donc nous partons du principe que Openclassrooms n'a pas mis à jour son exercice sinon on ne respecterait pas le RGPD sauf si les propriétaires listés dans le fichier ont renouvelés leur autorisation pour apparaître sur ce fichier .



Réglementation spécifique à l'immobilier

Origine ethnique ou prétendument raciale, Opinions politiques, Convictions religieuses ou philosophiques, L'appartenance syndicale, Données génétiques, biométriques, concernant la santé, concernant la vie sexuelle ou l'orientation sexuelle

5 - La création de la base de données



Création des clés secondaires ainsi que ses relations entre les tables

6 - Les requêtes demandées par la direction

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

```
1 • SELECT
2     COUNT(*) AS nb_ventes
3 FROM projet5.ventes
4 JOIN biens ON ventes.bien_id = biens.bien_id
5 WHERE date_mutation BETWEEN "2020-01-01" AND "2020-06-30" AND biens.type_local = "Appartement";
```

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

nb_ventes
31378

6 - Les requêtes demandées par la direction

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

```
1 WITH Appartement AS (  
2   SELECT *  
3   FROM biens WHERE biens.type_local = "Appartement"  
4 )  
5 SELECT  
6   regions.region_nom AS Régions,  
7   COUNT(vente_id) AS nb_ventes  
8 FROM projet5.ventes  
9 JOIN Appartement ON ventes.bien_id = Appartement.bien_id  
10 JOIN communes ON appartement.code_departement_code_commune = communes.code_departement_code_commune  
11 JOIN departements ON communes.departement_code = departements.departement_code  
12 JOIN regions ON departements.region_code = regions.region_code  
13 WHERE ventes.date_mutation BETWEEN "2020-01-01" AND "2020-06-30"  
14 GROUP BY regions.region_code, regions.region_nom  
15 ORDER BY nb_ventes DESC;
```

Result Grid | Filter Rows:

Export:



Wrap Cell Content:



Régions	nb_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

6 - Les requêtes demandées par la direction

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

```
1 WITH Appartement_vendus AS (  
2     SELECT  
3         COUNT(*) AS appartement_total  
4     FROM ventes  
5     JOIN biens ON ventes.bien_id = biens.bien_id  
6     WHERE biens.type_local = "Appartement"  
7 )  
8  
9     SELECT  
10        nombre_pieces_principales,  
11        COUNT(*) AS nb_ventes,  
12        round(COUNT(*) / (SELECT appartement_total FROM Appartement_vendus ) * 100 , 2) AS pourcentage  
13    FROM ventes  
14    JOIN biens ON ventes.bien_id = biens.bien_id  
15    WHERE biens.type_local = "Appartement"
```

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

	nombre_pieces_principales	nb_ventes	pourcentage
0		30	0.10
1		6739	21.48
2		9783	31.18
3		8966	28.57
4		4460	14.21
5		1114	3.55
6		204	0.65
7		54	0.17
8		17	0.05
9		8	0.03
10		2	0.01
11		1	0.00

6 - Les requêtes demandées par la direction

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

```
1 • SELECT
2     departements.departement_code AS "Nom du département",
3     round(AVG(ventes.valeur_fonciere / biens.surface_carrez),2) AS Prix_au_m2
4 FROM ventes
5 JOIN biens ON ventes.bien_id = biens.bien_id
6 JOIN communes ON biens.code_departement_code_commune = communes.code_departement_code_commune
7 JOIN departements ON communes.departement_code = departements.departement_code
8 GROUP BY departements.departement_code
9 ORDER BY Prix_au_m2 DESC
10 LIMIT 10;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: [IA](#)

Nom du département	Prix_au_m2
75	12045.35
92	7219.39
94	5340.51
06	4696.76
74	4667.13
93	4336.70
78	4225.25
69	4059.28
2A	4010.60
33	3764.14

6 - Les requêtes demandées par la direction

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

```
1 • SELECT
2     round(AVG(ventes.valeur_fonciere / biens.surface_carrez),2)
3     AS "Prix moyen du mètre carré d'une maison en Île-de-France"
4 FROM ventes
5 JOIN biens ON ventes.bien_id = biens.bien_id
6 JOIN communes ON biens.code_departement_code_commune = communes.code_departement_code_commune
7 JOIN departements ON communes.departement_code = departements.departement_code
8 JOIN regions ON departements.region_code = regions.region_code
9 WHERE biens.type_local = "Maison" AND regions.region_nom = "Île-de-France";
10
```

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

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

3745.09

6 - Les requêtes demandées par la direction

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

```
1 • SELECT
2     regions.region_nom,
3     biens.bien_id,
4     surface_carrez,
5     valeur_fonciere
6 FROM ventes
7 JOIN biens ON ventes.bien_id = biens.bien_id
8 JOIN communes ON biens.code_departement_code_commune = communes.code_departement_code_commune
9 JOIN departements ON communes.departement_code = departements.departement_code
10 JOIN regions ON departements.region_code = regions.region_code
11 WHERE type_local = 'Appartement'
12 ORDER BY valeur_fonciere DESC
13 LIMIT 10;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: [IA](#)

	region_nom	bien_id	surface_carrez	valeur_fonciere
•	Ile-de-France	1	9.10	9000000.00
	Ile-de-France	2	64.00	8600000.00
	Ile-de-France	3	20.55	8577713.00
	Ile-de-France	4	42.77	7620000.00
	Ile-de-France	5	253.30	7600000.00
	Ile-de-France	6	139.90	7535000.00
	Ile-de-France	7	360.95	7420000.00
	Ile-de-France	8	595.00	7200000.00
	Ile-de-France	9	122.56	7050000.00
	Ile-de-France	10	79.38	6600000.00

6 - Les requêtes demandées par la direction

7.

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

```
1 WITH ventes_T1 AS (  
2     SELECT  
3         COUNT(vente_id) AS ventes_premier_tri  
4     FROM ventes  
5     WHERE ventes.date_mutation BETWEEN "2020-01-01" AND "2020-03-31"  
6 ),  
7 ventes_T2 AS (  
8     SELECT  
9         COUNT(vente_id) AS ventes_second_tri  
10    FROM ventes  
11    WHERE ventes.date_mutation BETWEEN "2020-04-01" AND "2020-06-30"  
12 )  
13  
14 SELECT  
15     ventes_premier_tri,  
16     ventes_second_tri,  
17     ((ventes_second_tri - ventes_premier_tri) / ventes_premier_tri) *100 AS evolution  
18 FROM ventes_T1, ventes_T2;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: [IA](#)

ventes_premier_tri	ventes_second_tri	evolution
16776	17393	3.6779

6 - Les requêtes demandées par la direction

8.

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

```
1 • SELECT
2     regions.region_nom AS "Régions",
3     round(avg(ventes.valeur_fonciere / biens.surface_carrez),2) AS prix_au_metre_carré
4 FROM ventes
5 JOIN biens ON ventes.bien_id = biens.bien_id
6 JOIN communes ON biens.code_departement_code_commune = communes.code_departement_code_commune
7 JOIN departements ON communes.departement_code = departements.departement_code
8 JOIN regions ON departements.region_code = regions.region_code
9 WHERE biens.type_local = "Appartement" AND biens.nombre_pieces_principales > 4
10 GROUP BY regions.region_nom
11 ORDER BY prix_au_metre_carré DESC ;
12
```

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

Régions	prix_au_metre_carré
Ile-de-France	8757.64
La Réunion	3641.81
Provence-Alpes-Côte d'Azur	3587.65
Corse	3104.88
Auvergne-Rhône-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-Comté	1251.19
Martinique	573.48

6 - Les requêtes demandées par la direction

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

```
1 • SELECT
2     communes.nom_commune AS "Noms des communes",
3     Count(*) AS nombre_de_ventes
4 FROM ventes
5 JOIN biens ON ventes.bien_id = biens.bien_id
6 JOIN communes ON biens.code_departement_code_commune = communes.code_departement_code_commune
7 WHERE ventes.date_mutation BETWEEN "2020-01-01" AND "2020-03-31"
8 GROUP BY communes.nom_commune
9 HAVING Count(ventes.vente_id) >= 50
10 ORDER BY nombre_de_ventes DESC ;
```

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

Noms des communes	nombre_de_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
Grenoble	106
Paris 9e Arrondissement	106
Boulogne-Billancourt	99
Paris 13e Arrondissement	94
Paris 7e Arrondissement	87

6 - Les requêtes demandées par la direction

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

```
1 • WITH appartement_T2 AS (  
2     SELECT  
3         ROUND(AVG(v.valeur_fonciere / b.surface_carrez),2) AS T2_prix_moyen_m2  
4         FROM projet5.ventes v  
5         JOIN projet5.biens b ON b.bien_id = v.bien_id  
6         WHERE b.type_local = 'Appartement'  
7         AND b.nombre_pieces_principales = 2  
8     ),  
9     appartement_T3 AS (  
10        SELECT  
11            ROUND(AVG(v.valeur_fonciere / b.surface_carrez),2) AS T3_prix_moyen_m2  
12            FROM projet5.ventes v  
13            JOIN projet5.biens b ON b.bien_id = v.bien_id  
14            WHERE b.type_local = 'Appartement'  
15            AND b.nombre_pieces_principales = 3  
16        )  
17    SELECT  
18        T2_prix_moyen_m2,  
19        T3_prix_moyen_m2,  
20        ROUND((T3_prix_moyen_m2 - T2_prix_moyen_m2) / T2_prix_moyen_m2 * 100, 2) AS difference_pourcentage  
21    FROM appartement_T2, appartement_T3
```

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

T2_prix_moyen_m2	T3_prix_moyen_m2	difference_pourcentage
4903.56	4299.88	-12.31

6 - Les requêtes demandées par la direction

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

```
1 WITH classement_communes AS (  
2     SELECT  
3         communes.departement_code,  
4         communes.nom_commune AS nom_de_la_commune,  
5         AVG(ventes.valeur_fonciere) AS moyenne_valeur_fonciere,  
6         RANK() OVER (PARTITION BY communes.departement_code ORDER BY AVG(ventes.valeur_fonciere) DESC) AS rang_de_la_commune  
7     FROM ventes  
8     JOIN biens ON ventes.bien_id = biens.bien_id  
9     JOIN communes ON biens.code_departement_code_commune = communes.code_departement_code_commune  
10    JOIN departements ON communes.departement_code = departements.departement_code  
11    JOIN regions ON departements.region_code = regions.region_code  
12    WHERE communes.departement_code IN (6,13,33,59,69)  
13    GROUP BY communes.departement_code, communes.nom_commune  
14    ORDER BY communes.departement_code, rang_de_la_commune  
15 )  
16 SELECT *  
17 FROM classement_communes  
18 WHERE rang_de_la_commune <=3
```

departement_code	nom_de_la_commune	moyenne_valeur_fonciere	rang_de_la_commune
06	Saint-Jean-Cap-Ferrat	968750.000000	1
06	Eze	655000.000000	2
06	Mouans-Sartoux	476898.100000	3
13	Gignac-la-Nerthe	330000.000000	1
13	Saint-Savournin	314425.000000	2
13	Cassis	313416.875000	3
33	Lège-Cap-Ferret	549500.636364	1
33	Vayres	335000.000000	2
33	Arcachon	307435.927273	3
59	Bersée	433202.000000	1
59	Cysoing	408550.000000	2
59	Halluin	322250.000000	3
69	Ville-sur-Jarnioux	485300.000000	1
69	Lyon 2e Arrondissement	455217.266667	2
69	Lyon 6e Arrondissement	426968.250000	3

6 - Les requêtes demandées par la direction

12.

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

```
1 WITH communes_avec_plus_de_10000_habitants AS (  
2     SELECT DISTINCT communes.nom_commune  
3     FROM communes  
4     WHERE communes.population_totale > 10000  
5 ),  
6 transactions_par_habitant AS (  
7     SELECT  
8         communes.nom_commune, COUNT(*) AS nombre_de_transactions, communes.population_totale,  
9         (COUNT(*) / communes.population_totale) * 1000 AS transactions_par_1000_habitants  
10    FROM ventes  
11    JOIN biens ON ventes.bien_id = biens.bien_id  
12    JOIN communes ON biens.code_departement_code_commune = communes.code_departement_code_commune  
13    WHERE communes.nom_commune IN (SELECT nom_commune FROM communes_avec_plus_de_10000_habitants)  
14    GROUP BY communes.nom_commune, communes.population_totale  
15 )  
16 SELECT  
17     nom_commune, transactions_par_1000_habitants  
18 FROM transactions_par_habitant  
19 ORDER BY transactions_par_1000_habitants DESC  
20 LIMIT 20;
```

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

nom_commune	transactions_par_1000_habitants
Paris 2e Arrondissement	5.8431
Paris 1er Arrondissement	4.9206
Paris 3e Arrondissement	4.6931
Arcachon	4.6226
La Baule-Escoubac	4.5842
Paris 4e Arrondissement	4.0830
Roquebrune-Cap-Martin	3.9874
Paris 8e Arrondissement	3.8345
Sanary-sur-Mer	3.4965
Paris 9e Arrondissement	3.4344
La Londe-les-Maures	3.4336
Paris 6e Arrondissement	3.3762
Saint-Cyr-sur-Mer	3.2409
Chantilly	3.1312
Pornichet	3.0594
Saint-Mandé	3.0563
Paris 10e Arrondissement	3.0393
Menton	2.9373
Saint-Hilaire-de-Riez	2.8693