

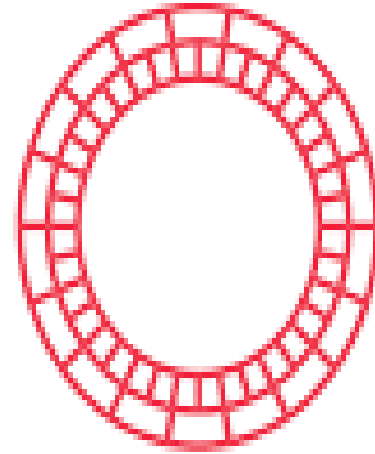
PROJET 3: Créez et utilisez une base de données immobilière avec SQL

Soutenance de projet
11 MAI 2023

Marie NGAHA



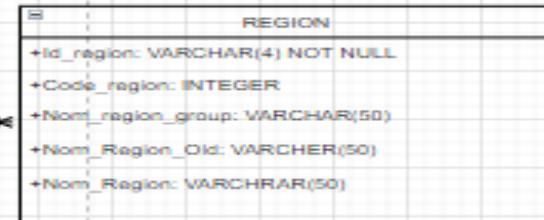
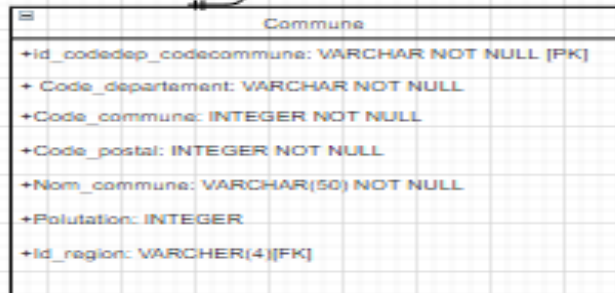
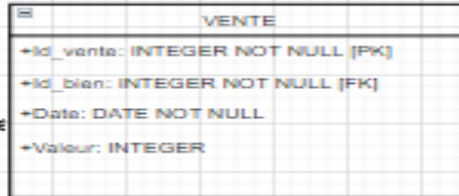
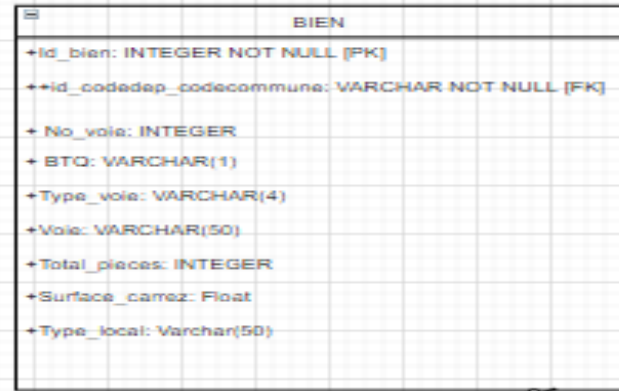
- Schémas relationnel
- Base de données
- Requêtes



LA PLACE IMMO

AGENCE IMMOBILIERE

Schémas relationnel



1

1

1

1

Schémas relationnel

- 4 tables:
 - Bien**: Qui contient toutes les informations concernant un bien, comme le type de bien, sa superficie, son nombre de pièce, le type de voie ,....
 - Commune**: Qui contient toutes les informations qui concerne l'adresse d'un bien, comme le nom de la commune, la population ,son code,....
 - Vente**: Qui contient toutes les information concernant la vente d'un bien, comme ,la date et sa valeur.
 - Region**: contient le nom de la region et ses identifiants....

Ces 4 tables ont chacune une clé primaire avec un numéro unique. Ces 4 tables sont aussi reliées par trois clés étrangères qui relient bien à vente et ensuite bien à commune puis commune à region.

Base de donnée avec les données importés sur Mysql

- table region

✓ Affichage des lignes 0 - 19 (total de 20, traitement en 0,0014 seconde(s).)

[SELECT](#) * FROM `region`

☐ Profilage [\[Éditer en ligne \]](#) [\[Éditer \]](#) [\[Expliquer SQL \]](#) [\[Créer le code source PHP \]](#) [\[Actualiser \]](#)

☐ Tout afficher | Nombre de lignes : 25 | Filtrer les lignes: Chercher dans cette table | Trier par clé : Aucun(e)

+ Options

	Id_region	Code_region	Nom_region_group	Nom_region_old	Nom_region
<input type="checkbox"/> Éditer Copier Supprimer	Id_region	0	Nom_region_group	Region_nom_old	Nom_region
<input type="checkbox"/> Éditer Copier Supprimer	R00	0	DROM-COM	Collectivités d'outre-mer	Collectivités d'outre-mer
<input type="checkbox"/> Éditer Copier Supprimer	R01	1	DROM-COM	Guadeloupe	Guadeloupe
<input type="checkbox"/> Éditer Copier Supprimer	R02	2	DROM-COM	Martinique	Martinique
<input type="checkbox"/> Éditer Copier Supprimer	R03	3	DROM-COM	Guyane	Guyane
<input type="checkbox"/> Éditer Copier Supprimer	R04	4	DROM-COM	La Réunion	La Réunion
<input type="checkbox"/> Éditer Copier Supprimer	R06	6	DROM-COM	Mayotte	Mayotte
<input type="checkbox"/> Éditer Copier Supprimer	R11	11	Ile-de-France	Ile-de-France	Ile-de-France
<input type="checkbox"/> Éditer Copier Supprimer	R24	24	Province	Centre-Val de Loire	Centre-Val de Loire
<input type="checkbox"/> Éditer Copier Supprimer	R27	27	Province	Bourgogne	Bourgogne-Franche-Comté

table commune

✓ Affichage des lignes 0 - 24 (total de 3126, traitement en 0,0012 seconde(s).)

[SELECT](#) * FROM `commune`

☐ Profilage [\[Éditer en ligne \]](#) [\[Éditer \]](#) [\[Expliquer SQL \]](#) [\[Créer le code source PHP \]](#) [\[Actualiser \]](#)

1 | > >> | Nombre de lignes : 25 | Filtrer les lignes: Chercher dans cette table | Trier par clé : Aucun(e)

+ Options

	Id_codedep_codecommune	Code_departement	Code_commune	Code_postal	Nom_Commune	Population	Id_region
<input type="checkbox"/> Éditer Copier Supprimer	01_004	01	4	1500	AMBERIEU-EN-BUGEY	244	R28
<input type="checkbox"/> Éditer Copier Supprimer	01_014	01	14	1100	ARBENT	760	R84
<input type="checkbox"/> Éditer Copier Supprimer	01_030	01	30	1480	BEAUREGARD	104	R84
<input type="checkbox"/> Éditer Copier Supprimer	01_031	01	31	1100	BELLIGNAT	137	R24
<input type="checkbox"/> Éditer Copier Supprimer	01_033	01	33	1200	VALSERHONNE	1522	R32
<input type="checkbox"/> Éditer Copier Supprimer	01_034	01	34	1300	BELLEY	383	R84
<input type="checkbox"/> Éditer Copier Supprimer	01_060	01	60	1110	BRENOD	1087	R27
<input type="checkbox"/> Éditer Copier Supprimer	01_071	01	71	1170	CESSY	123	R44
<input type="checkbox"/> Éditer Copier Supprimer	01_079	01	79	1260	CHAMPAGNE-EN-VAUX	1972	R24

Console de requêtes SQL | [Sélectionner](#) | [Options](#) | [Historique](#) | [Effacer](#)

Base de donnée avec les données importés sur Mysql

- Table bien

✓ Affichage des lignes 0 - 24 (total de 34170, traitement en 0,0189 seconde(s))

SELECT * FROM `bien`

☐ Profilage [Éditer en ligne] [Éditer] [Expliquer SQL] [Créer le code source PHP] [Actualiser]

1 > >> Nombre de lignes : 25 Filtrer les lignes: Chercher dans cette table Trier par clé : Aucun(e)

+ Options

	Id_bien	No_voie	BTQ	Type_voie	Voie	Total_pieces	Surface_local	Surface_carrez	Type_local	Id_codedep_cod
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer	0	0	BTQ	Type_voie	Voie	0	0	0	Type_local	Id_codedep_cod
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer	1	347	RUE		DU CHATEAU	3	48	48	Appartement	01_103
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer	2	4	BD		EDOUARD BAUDOUIN	1	40	39	Appartement	06_004
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer	3	20	B	RUE	MARCEAU	3	82	80	Appartement	06_088
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer	4	550	RTE		DES VESPINS RN7	1	27	27	Appartement	06_123
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer	5	9300	RES		LES ARPEGES BD DES ABA	2	47	47	Appartement	13_005
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer	6	27	RUE		DU GRAND MADIER	1	24	25	Appartement	13_028

Table vente

✓ Affichage des lignes 0 - 24 (total de 34170, traitement en 0,0727 seconde(s))

SELECT * FROM `vente`

☐ Profilage [Éditer en ligne] [Éditer] [Expliquer SQL] [Créer le code source PHP] [Actualiser]

1 > >> Nombre de lignes : 25 Filtrer les lignes: Chercher dans cette table Trier par clé : Aucun(e)

+ Options

	Id_vente	Da_te	Valeur	Id_bien
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer	0	0000-00-00	0	0
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer	1	2020-01-02	165000	1
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer	2	2020-01-02	355680	2
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer	3	2020-01-02	229500	3
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer	4	2020-01-02	125000	4
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer	5	2020-01-02	90000	5
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer	6	2020-01-02	93000	6
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer	7	2020-01-02	298100	7
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer	8	2020-01-02	163500	8
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer	9	2020-01-02	52000	9

REQUETTES

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

- SELECT COUNT(Type_local) AS nombre_total_de_ventes
- FROM vente
- JOIN bien ON vente.Id_bien = bien.Id_bien
- WHERE vente.Da_te BETWEEN '2020-01-01' AND '2020-06-30'
- AND bien.Type_local = 'Appartement';

La requête SQL a été exécutée avec succès.

```
SELECT COUNT(Type_local) AS nombre_total_de_ventes FROM vente JOIN bien ON vente.Id_bien = bien.Id_bien WHERE vente.Da_te BETWEEN '2020-01-01' AND '2020-06-30' AND bien.Type_local = 'Appartement';
```

☐ Profilage [[Éditer en ligne](#)] [[Éditer](#)] [[Expliquer SQL](#)] [[Créer le code source PHP](#)] [[Actualiser](#)]

+ Options

nombre_total_de_ventes

31378

- Cette requête utilise une jointure pour relier les tables vente et bien afin d'obtenir les informations nécessaires pour calculer le nombre total de ventes d'appartements au 1er semestre 2020. Elle inclut également une condition pour ne sélectionner que les ventes d'appartements en utilisant la colonne Type_local de la table bien.

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

- SELECT region.Nom_region, COUNT(Type_local) AS nombre_de_ventes
- FROM vente
- JOIN bien ON vente.Id_bien = bien.Id_bien
- JOIN commune ON bien.Id_codedep_codecommune = commune.Id_codedep_codecommune
- JOIN region ON commune.Id_region = region.Id_region
- WHERE vente.Date BETWEEN '2020-01-01' AND '2020-06-30'
- AND bien.Type_local = 'Appartement'
- GROUP BY region.Nom_region;

☐ Tout afficher | Nombre de lignes : 25 ▼ | Filtrer les lig

+ Options

Nom_region	nombre_de_ventes_Appartements
Auvergne-Rhône-Alpes	17132
Bourgogne-Franche-Comté	538
Bretagne	202
Centre-Val de Loire	295
Grand Est	1567
Hauts-de-France	6782
Ile-de-France	23
Normandie	907
Nouvelle-Aquitaine	886
Occitanie	1414
Pays de la Loire	93
Provence-Alpes-Côte d'Azur	1539

- Cette requête utilise des jointures pour relier les tables vente, bien, commune et region afin d'obtenir les informations nécessaires pour calculer le nombre de ventes d'appartements par région pour le 1er semestre 2020. Elle inclut également une condition pour ne sélectionner que les ventes d'appartements en utilisant la colonne Type_local de la table bien.

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

- SET @total = (SELECT COUNT(Type_local)
- FROM bien
- WHERE Type_local = "Appartement");
- SELECT Total_pieces,
- COUNT(Type_local) AS appartements,
- ROUND(COUNT(Type_local)/@total*100,2)
- AS 'proportion %'
- FROM bien
- WHERE Type_local = "Appartement"
- GROUP BY Total_pieces
- ORDER BY Total_pieces;

☐ Profilage [[Éditer en ligne](#)] [[Éditer](#)] [[Expliquer SQL](#)]

☐ Tout afficher | Nombre de lignes : 25 ▼

+ Options

Total_pieces	appartements	proportion %
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

- Cette requête calcule le nombre de ventes pour chaque nombre de pièces dans les appartements et divise ce nombre par le nombre total de ventes d'appartements pour obtenir la proportion.

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

- SELECT Code_departement, AVG(Valeur / Surface_carrez) AS Prix_m2
- FROM vente
- JOIN bien ON vente.Id_bien = bien.Id_bien
- JOIN commune ON bien.Id_codedep_codecommune = commune.Id_codedep_codecommune
- WHERE Surface_carrez > 0
- GROUP BY Code_departement
- ORDER BY Prix_m2 DESC
- LIMIT 10;

+ Options	
Code_departement	Prix_m2 ▾ 1
75	12084.094011350948
92	7300.659801376434
94	5427.8658761096085
74	4781.394537302809
06	4755.507115076039
93	4385.746206169728
78	4275.575227795183
69	4100.283052626802
2A	4062.9127852010083
33	3807.235671550534

- Cette requête calcule le prix moyen au mètre carré pour chaque département en divisant la valeur de vente par la surface Carrez de chaque bien vendu. Elle ne prend en compte que les biens dont la surface Carrez est supérieure à zéro. Ensuite, elle trie les résultats par ordre décroissant de prix au mètre carré et ne renvoie que les 10 premiers résultats.

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

- SELECT AVG(Valeur / Surface_carrez) AS Prix_m2
- FROM vente
- JOIN bien ON vente.Id_bien = bien.Id_bien
- JOIN commune ON bien.Id_codedep_codecommune = commune.Id_codedep_codecommune
- JOIN region ON commune.Id_region = region.Id_region
- WHERE Surface_carrez > 0 AND Type_local = 'Maison' AND Nom_region = 'Île-de-France';

+ Options

Prix_m2

3764.8427202559565

- Cette requête calcule le prix moyen au mètre carré pour les maisons en Île-de-France en divisant la valeur de vente par la surface Carrez de chaque bien vendu. Elle ne prend en compte que les biens dont la surface Carrez est supérieure à zéro et qui sont des maisons situées en Île-de-France

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

- SELECT Nom_region, Surface_carrez, Valeur, code_departement, Type_local
- FROM vente
- JOIN bien ON vente.Id_bien = bien.Id_bien
- JOIN commune ON bien.Id_codedep_codecommune = commune.Id_codedep_codecommune
- JOIN region ON commune.Id_region = region.Id_region
- WHERE Type_local = 'Appartement'
- ORDER BY Valeur DESC
- LIMIT 10;

+ Options

Nom_region	Surface_carrez	Valeur ▾ 1	code_departement	Type_local
Auvergne-Rhône-Alpes	9	9000000	75	Appartement
Auvergne-Rhône-Alpes	64	8600000	91	Appartement
Auvergne-Rhône-Alpes	20	8577713	75	Appartement
Auvergne-Rhône-Alpes	42	7620000	75	Appartement
Auvergne-Rhône-Alpes	253	7600000	75	Appartement
Hauts-de-France	139	7535000	75	Appartement
Auvergne-Rhône-Alpes	360	7420000	75	Appartement
Auvergne-Rhône-Alpes	595	7200000	75	Appartement
Hauts-de-France	122	7050000	75	Appartement
Hauts-de-France	79	6600000	75	Appartement

- la requête commence par sélectionner les colonnes Nom_region, Surface_carrez et Valeur à partir de la table vente.
- Ensuite, elle joint la table bien à la table vente en utilisant la colonne Id_bien pour faire correspondre les lignes des deux tables.
- Elle joint également la table commune à la table bien en utilisant la colonne Id_codedep_codecommune pour faire correspondre les lignes des deux tables.
- Enfin, elle joint la table region à la table commune en utilisant la colonne Id_region pour faire correspondre les lignes des deux tables. La requête ajoute une condition pour ne sélectionner que les lignes où la colonne Type_local est égale à 'Appartement', c'est-à-dire uniquement les appartements.
- Elle trie ensuite les résultats par ordre décroissant de valeur de vente en utilisant la clause ORDER BY Valeur DESC.
- Enfin, elle ne renvoie que les 10 premiers résultats en utilisant la clause LIMIT 10.

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

- WITH VentesT1 AS (
 - SELECT COUNT(*) AS NbVentes
 - FROM vente
 - WHERE Da_te BETWEEN '2020-01-01' AND '2020-03-31'
-),
- VentesT2 AS (
 - SELECT COUNT(*) AS NbVentes
 - FROM vente
 - WHERE Da_te BETWEEN '2020-04-01' AND '2020-06-30'
-)
- SELECT (VentesT2.NbVentes - VentesT1.NbVentes) * 100.0 / VentesT1.NbVentes AS TauxEvolution
- FROM VentesT1, VentesT2;

+ Options

TauxEvolution

3.67787

- Cette requête utilise des sous-requêtes pour calculer le nombre de ventes au premier et au second trimestre de 2020. Elle soustrait ensuite le nombre de ventes au premier trimestre du nombre de ventes au second trimestre et divise le résultat par le nombre de ventes au premier trimestre pour obtenir le taux d'évolution en pourcentage

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

- SELECT Nom_region, Total_pieces, Type_local, AVG(Valeur / Surface_carrez) AS Prix_m2
- FROM vente
- JOIN bien ON vente.Id_bien = bien.Id_bien
- JOIN commune ON bien.Id_codedep_codecommune = commune.Id_codedep_codecommune
- JOIN region ON commune.Id_region = region.Id_region
- WHERE Type_local = 'Appartement' AND Total_pieces > 4 AND Surface_carrez > 0
- GROUP BY Nom_region
- ORDER BY Prix_m2 DESC;

Nom_region	Total_pieces	Type_local	Prix_m2 ▾ 1
Auvergne-Rhône-Alpes	5	Appartement	7084.81615938598
Hauts-de-France	5	Appartement	4276.23110487189
Provence-Alpes-Côte d'Azur	5	Appartement	3298.784495741247
Occitanie	5	Appartement	2635.5723361383825
Grand Est	5	Appartement	2601.402744289155
Centre-Val de Loire	5	Appartement	2323.2965137104834
Bourgogne-Franche-Comté	6	Appartement	1699.6413641408963
Normandie	6	Appartement	1690.2416776586297
Bretagne	6	Appartement	1594.1497175027505
Nouvelle-Aquitaine	5	Appartement	1376.2035100637768
Ile-de-France	6	Appartement	1029.3012927132415

- La requête commence par sélectionner les colonnes Nom_region et AVG(Valeur / Surface_carrez) à partir de la table vente. La fonction d'agrégation AVG() calcule la moyenne des valeurs de vente divisées par la surface Carrez pour chaque région.
- Ensuite, elle joint la table bien à la table vente en utilisant la colonne Id_bien pour faire correspondre les lignes des deux tables.
- Elle joint également la table commune à la table bien en utilisant la colonne Id_codedep_codecommune pour faire correspondre les lignes des deux tables.
- Enfin, elle joint la table region à la table commune en utilisant la colonne Id_region pour faire correspondre les lignes des deux tables. La requête ajoute plusieurs conditions pour ne sélectionner que les lignes où la colonne Type_local est égale à 'Appartement', c'est-à-dire uniquement les appartements, où la colonne Total_pieces est supérieure à 4, c'est-à-dire uniquement les appartements de plus de 4 pièces, et où la colonne Surface_carrez est supérieure à zéro.
- Elle groupe ensuite les résultats par région en utilisant la clause GROUP BY Nom_region.
- Elle trie ensuite les résultats par ordre décroissant de prix au mètre carré en utilisant la clause ORDER BY Prix_m2 DESC.

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

- SELECT Nom_Commune, Valeur
- FROM vente
- JOIN bien ON vente.Id_bien = bien.Id_bien
- JOIN commune ON bien.Id_codedep_codecommune = commune.Id_codedep_codecommune
- WHERE Da_te BETWEEN '2020-01-01' AND '2020-03-31'
- GROUP BY Nom_Commune
- HAVING COUNT(*) >= 50;

1	>	>>	<input type="checkbox"/> Tout afficher	Nombre de lignes :	25	Fi
+ Options						
Nom_Commune	Valeur					
AJACCIO	213390					
ANGERS	105003					
ANTIBES	355680					
ASNIERES-SUR-SEINE	435000					
BORDEAUX	86025					
BOULOGNE-BILLANCOURT	548000					
COURBEVOIE	395300					
GRENOBLE	149500					
ISSY-LES-MOULINEAUX	1550000					
LA CIOTAT	93000					
LEVALLOIS-PERRET	406400					
LILLE	585000					
MARSEILLE 1ER	146000					
MARSEILLE 4EME	69350					
MARSEILLE 8EME	298100					
MARSEILLE 9EME	70000					
MONTREUIL	572000					

- Cette requête sélectionne les ventes qui ont eu lieu au premier trimestre de 2020 en utilisant la condition `Da_te BETWEEN '2020-01-01' AND '2020-03-31'`. Elle joint ensuite les tables `bien` et `commune` pour récupérer le nom de la commune pour chaque vente. Ensuite, elle groupe les résultats par commune en utilisant la clause `GROUP BY Nom_Commune` et ne renvoie que les communes ayant eu au moins 50 ventes en utilisant la clause `HAVING COUNT(*) >= 50`

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

- WITH PrixM2_2P AS (
 - SELECT AVG(Valeur / Surface_carrez) AS Prix_m2
 - FROM vente
 - JOIN bien ON vente.Id_bien = bien.Id_bien
 - WHERE Type_local = 'Appartement' AND Total_pieces = 2 AND Surface_carrez > 0
-),
- PrixM2_3P AS (
 - SELECT AVG(Valeur / Surface_carrez) AS Prix_m2
 - FROM vente
 - JOIN bien ON vente.Id_bien = bien.Id_bien
 - WHERE Type_local = 'Appartement' AND Total_pieces = 3 AND Surface_carrez > 0
-)
- SELECT (PrixM2_3P.Prix_m2 - PrixM2_2P.Prix_m2) * 100.0 / PrixM2_2P.Prix_m2 AS DiffPourcentage
- FROM PrixM2_2P, PrixM2_3P;

+ Options

DiffPourcentage

-12.68249373588277

- La requête commence par définir deux sous-requêtes nommées PrixM2_2P et PrixM2_3P à l'aide de la clause WITH.
- La première sous-requête PrixM2_2P calcule le prix moyen au mètre carré pour les appartements de 2 pièces en sélectionnant la colonne AVG(Valeur / Surface_carrez) à partir de la table vente. La fonction d'agrégation AVG() calcule la moyenne des valeurs de vente divisées par la surface Carrez pour chaque appartement de 2 pièces.
- Cette sous-requête joint également la table bien à la table vente en utilisant la colonne Id_bien pour faire correspondre les lignes des deux tables. Elle ajoute plusieurs conditions pour ne sélectionner que les lignes où la colonne Type_local est égale à 'Appartement', c'est-à-dire uniquement les appartements, où la colonne Total_pieces est égale à 2, c'est-à-dire uniquement les appartements de 2 pièces, et où la colonne Surface_carrez est supérieure à zéro.
- La seconde sous-requête PrixM2_3P fonctionne de manière similaire, mais calcule le prix moyen au mètre carré pour les appartements de 3 pièces en sélectionnant uniquement les lignes où la colonne Total_pieces est égale à 3. Ensuite, la requête principale sélectionne la différence entre le prix moyen au mètre carré des appartements de 3 pièces et le prix moyen au mètre carré des appartements de 2 pièces, multipliée par 100 et divisée par le prix moyen au mètre carré des appartements de 2 pièces pour obtenir la différence en pourcentage.
- Elle utilise les résultats des deux sous-requêtes pour effectuer ce calcul en utilisant les alias PrixM2_3P.Prix_m2 et PrixM2_2P.Prix_m2.

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

- WITH CommunesTop3 AS (
 - SELECT Code_departement, Nom_Commune, AVG(Valeur) AS ValeurMoyenne
 - FROM vente
 - JOIN bien ON vente.Id_bien = bien.Id_bien
 - JOIN commune ON bien.Id_codedep_codecommune = commune.Id_codedep_codecommune
 - WHERE Code_departement IN ('06', '13', '33', '59', '69')
 - GROUP BY Code_departement, Nom_Commune
 - ORDER BY Code_departement, ValeurMoyenne DESC
-)SELECT Code_departement, Nom_Commune, ValeurMoyenne
- FROM (
 - SELECT Code_departement, Nom_Commune, ValeurMoyenne,
 - ROW_NUMBER() OVER (PARTITION BY Code_departement ORDER BY ValeurMoyenne DESC) AS Rang
- FROM CommunesTop3
-) AS CommunesRang
- WHERE Rang <= 3;

+ Options

Code_departement	Nom_Commune	ValeurMoyenne
06	SAINT-JEAN-CAP-FERRAT	968750.0000
06	EZE	655000.0000
06	MOUANS-SARTOUX	476898.0000
13	GIGNAC-LA-NERTHE	330000.0000
13	SAINT SAVOURNIN	314425.0000
13	CASSIS	313416.8750
33	LEGE-CAP-FERRET	549500.6364
33	VAYRES	335000.0000
33	ARCACHON	307435.9273
59	BERSEE	433202.0000
59	CYSOING	408550.0000
59	HALLUIN	322250.0000
69	VILLE SUR JARNIOUX	485300.0000
69	LYON 2EME	455217.2609
69	LYON 6EME	426968.2500

- Cette requête utilise une sous-requête pour calculer la valeur foncière moyenne pour chaque commune des départements 6, 13, 33, 59 et 69. Elle trie ensuite les résultats par département et par ordre décroissant de valeur foncière moyenne. Ensuite, elle utilise la fonction de fenêtre ROW_NUMBER() pour attribuer un rang à chaque commune en fonction de sa valeur foncière moyenne. Enfin, elle ne renvoie que les communes ayant un rang inférieur ou égal à 3 pour chaque département.
- La requête commence par définir une sous-requête nommée CommunesTop3 à l'aide de la clause WITH.
- Cette sous-requête calcule la valeur foncière moyenne pour chaque commune des départements 6, 13, 33, 59 et 69 en sélectionnant les colonnes Code_departement, Nom_Commune et AVG(Valeur) à partir de la table vente. La fonction d'agrégation AVG() calcule la moyenne des valeurs de vente pour chaque commune.
- Cette sous-requête joint également les tables bien et commune à la table vente en utilisant les colonnes Id_bien et Id_codedep_codecommune pour faire correspondre les lignes des trois tables.
- Elle ajoute une condition pour ne sélectionner que les lignes où la colonne Code_departement est dans la liste ('06', '13', '33', '59', '69'), c'est-à-dire uniquement les communes des départements 6, 13, 33, 59 et 69. Elle groupe ensuite les résultats par département et par commune en utilisant la clause GROUP BY Code_departement, Nom_Commune.
- Elle trie ensuite les résultats par département et par ordre décroissant de valeur foncière moyenne en utilisant la clause ORDER BY Code_departement, ValeurMoyenne DESC.
- Ensuite, la requête principale sélectionne les colonnes Code_departement, Nom_Commune et ValeurMoyenne à partir du résultat de la sous-requête.
- Elle utilise également la fonction de fenêtre ROW_NUMBER() pour attribuer un rang à chaque commune en fonction de sa valeur foncière moyenne. Cette fonction est appliquée sur chaque groupe de communes appartenant au même département grâce à la clause PARTITION BY Code_departement.
- Enfin, elle ne renvoie que les communes ayant un rang inférieur ou égal à 3 en utilisant la condition WHERE Rang <= 3.

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

- `SELECT Nom_Commune, COUNT(*) *
1000.0 / Population AS
TransactionsPar1000Hab`
- `FROM vente`
- `JOIN bien ON vente.Id_bien = bien.Id_bien`
- `JOIN commune ON
bien.Id_codedep_codecommune =
commune.Id_codedep_codecommune`
- `WHERE Population > 10000`
- `GROUP BY Nom_Commune`
- `ORDER BY TransactionsPar1000Hab DESC`
- `LIMIT 20;`

+ Options	
Nom_Commune	TransactionsPar1000Hab ▾ 1
NICE	27.07730
REIMS	10.52946
DAX	5.00540
THIAIS	3.74392
EPERNAY	2.15208
CROIX	1.57654
L'HAY-LES-ROSES	1.52489
VILLARD-DE-LANS	1.48280
PERPIGNAN	1.36153
BOURG-LA-REINE	1.35165
CASSIS	1.13130
NIORT	0.91823
VILLEPINTE	0.80441
MARTIGNAS SUR JALLE	0.60827
ALES	0.58881
LES HOUCHES	0.58514
ESSEY LES NANCY	0.42608
LA VILLE DIEU DU TEMPLE	0.38937
GRISY-SUISNES	0.33645
SARTENE	0.24788

- Cette requête calcule le nombre de transactions pour chaque commune en utilisant la fonction d'agrégation COUNT(*). Elle multiplie ensuite ce nombre par 1000 et le divise par la population de la commune pour obtenir le nombre de transactions pour 1000 habitants. Elle ne prend en compte que les communes dont la population est supérieure à 10 000 habitants. Ensuite, elle trie les résultats par ordre décroissant de transactions pour 1000 habitants et ne renvoie que les 20 premiers résultats.

