



Parcours
Data Analyst
V2

OPENCLASSROOMS

PROJET P3 - Créez et utilisez une base de données immobilière avec SQL

Résultats des Requêtes avec alternatives de codes

Version	Auteur	Description
092022	Adeline Le Ray	Première émission



Table des matières

1. Objectif du document.....	4
2. Requête n°1 - Nombre total d' appartements vendus au 1er semestre 2020.....	5
a. Jointure avec WHERE	5
b. Jointure avec JOIN	6
3. Requête n°2 - Le nombre de ventes d' appartement par région pour le 1er semestre 2020	7
a. Jointure avec WHERE	7
b. Jointure avec JOIN	8
4. Requête n°3 - Proportion des ventes d' appartements par le nombre de pièces	9
a. Jointure avec WHERE	9
b. Jointure avec JOIN	10
5. Requête n°4 - Liste des 10 départements où le prix du mètre carré est le plus élevé.....	11
a. Jointure avec WHERE	11
b. Jointure avec JOIN	12
6. Requête n°5 - Prix moyen du mètre carré d' une maison en Île-de-France.	13
a. WHERE et GROUP BY	13
b. JOIN, WHERE et GROUP BY	14
7. Requête n°6 - Liste des 10 appartements les plus chers avec la région et le nombre de mètres carrés	15
a. WHERE et ORDER BY	15
b. JOIN et ORDER BY	16
8. Requête n°7 - Taux d' évolution du nombre de ventes entre le premier et le second trimestre de 2020	17



a.	CASE WHEN	17
b.	WITH.....	18
9.	Requête n°8 - Le classement des régions par rapport au prix au mètre carré des appartements de plus de 4 pièces.....	19
a.	WHERE, GROUP BY	19
b.	JOIN, GROUP BY	20
10.	Requête n°9 - Liste des communes ayant eu au moins 50 ventes au 1er trimestre	21
a.	WHERE et HAVING	21
b.	JOIN et HAVING	23
11.	Requête n°10 - Différence en pourcentage du prix au mètre carré entre un appartement de 2 pièces et un appartement de 3 pièces	24
a.	CREATE TEMPORARY TABLE	24
b.	WITH.....	25
12.	Requête n°11 - Les moyennes de valeurs foncières pour le top 3 des communes des départements 6, 13, 33, 59 et 69	26
a.	WITH, RANK() OVER, WHERE	26
b.	WITH, RANK () OVER, JOIN, UNION	27



1. Objectif du document

L'objectif de ce document est de présenter les requêtes et leurs résultats.

Les données utilisées pour les requêtes sont celles du premier semestre 2020 uniquement pour servir de POC (Proof of Concept).

Les données à extraire via des requêtes SQL sur les bases de données sont les suivantes :

1. Nombre total d'appartements vendus au 1er semestre 2020.
2. Le nombre de ventes d'appartement par région pour le 1er semestre 2020.
3. Proportion des ventes d'appartements par le nombre de pièces.
4. Liste des 10 départements où le prix du mètre carré est le plus élevé.
5. Prix moyen du mètre carré d'une maison en Île-de-France.
6. Liste des 10 appartements les plus chers avec la région et le nombre de mètres carrés.
7. Taux d'évolution du nombre de ventes entre le premier et le second trimestre de 2020
8. Le classement des régions par rapport au prix au mètre carré des appartement de plus de 4 pièces.
9. Liste des communes ayant eu au moins 50 ventes au 1er trimestre
10. Différence en pourcentage du prix au mètre carré entre un appartement de 2 pièces et un appartement de 3 pièces.
11. Les moyennes de valeurs foncières pour le top 3 des communes des départements 6, 13, 33, 59 et 69

Pour certaines requêtes, des variantes de code sont indiquées afin de présenter les différentes approches étudiées lors du projet P3.



Parcours
Data Analyst
V2

OPENCLASSROOMS

2. Requête n°1 - Nombre total d' appartements vendus au 1er semestre 2020.

a. Jointure avec WHERE

Requete1a

```
1 • SELECT b.TYPE_LOCAL, count(*) as nb_ventes_1er_semestre
2 FROM dan_p3.bien b, dan_p3.vente v
3 WHERE
4     v.DATE_VENTE BETWEEN '2020-01-01' and '2020-06-30'
5     AND b.ID_BIEN = v.ID_BIEN
6     AND lower(b.TYPE_LOCAL) = 'appartement'
7 GROUP BY b.TYPE_LOCAL
8 ;
```

Result Grid

	TYPE_LOCAL	nb_ventes_1er_semestre
▶	Appartement	31326

Result 1

Output

Action Output

#	Time	Action	Message
✓ 1	09:15:38	SELECT b.TYPE_LOCAL, count(*) as nb_ventes_1er_semestre FROM dan_p3.bien b, dan_p3.vente v...	1 row(s) returned



Parcours
Data Analyst
V2

OPENCLASSROOMS

b. Jointure avec JOIN

Requete1b x

```
1 • SELECT TYPE_LOCAL, count(*) as nb_vente_1er_semestre
2 FROM dan_p3.bien
3 JOIN dan_p3.vente USING (ID_BIEN)
4 WHERE
5     DATE_VENTE BETWEEN '2020-01-01' AND '2020-06-30'
6     AND TYPE_LOCAL = 'appartement'
7 GROUP BY TYPE_LOCAL
8 ;
```

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

	TYPE_LOCAL	nb_vente_1er_semestre
▶	Appartement	31326

Result 1 x

Output

Action Output

#	Time	Action	Message
✓ 1	09:17:25	SELECT TYPE_LOCAL, count(*) as nb_vente_1er_semestre FROM dan_p3.bien JOIN dan_p3.vente ...	1 row(s) returned



Parcours
Data Analyst
V2

OPENCLASSROOMS

3. Requête n°2 - Le nombre de ventes d' appartement par région pour le 1er semestre 2020

a. Jointure avec WHERE

Requete2a*

```
1 • SELECT b.TYPE_LOCAL as Type_de_local, r.REG as Region, count(*) as Nb_vente_1er_semestre
2 FROM dan_p3.bien b, dan_p3.commune c, dan_p3.region r, dan_p3.vente v
3 WHERE
4     v.DATE_VENTE BETWEEN '2020-01-01' AND '2020-06-30'
5     AND b.ID_BIEN = v.ID_BIEN
6     AND b.ID_COM = c.ID_COM
7     AND c.ID_REG = r.ID_REG
8     AND b.TYPE_LOCAL = 'appartement'
9 GROUP BY b.TYPE_LOCAL, r.REG
10 ORDER BY Nb_vente_1er_semestre DESC;
```

Result Grid

Type_de_local	Region	Nb_vente_1er_semestre
Appartement	Île-de-France	13974
Appartement	Provence-Alpes-Côte d'Azur	3639
Appartement	Auvergne-Rhône-Alpes	3248
Appartement	Nouvelle-Aquitaine	1931
Appartement	Occitanie	1640
Appartement	Pays de la Loire	1356
Appartement	Hauts-de-France	1251
Appartement	Grand Est	982
Appartement	Bretagne	982
Appartement	Normandie	860
Appartement	Centre-Val de Loire	693
Appartement	Bourgogne-Franche-Comté	376
Appartement	Corse	222
Appartement	Martinique	94
Appartement	La Réunion	44
Appartement	Guyane	34

Result 2

Output

Action Output

#	Time	Action	Message
1	11:00:02	SELECT b.TYPE_LOCAL as Tvoe de local.r.REG as Region.count(*) as Nb_vente_1er semestre F...	16 row(s) returned



Parcours
Data Analyst
V2

OPENCLASSROOMS

b. Jointure avec JOIN

Requete2b*

```
1 • SELECT TYPE_LOCAL as Type_de_local, REG as Region, count(*) as Nb_vente_1er_semestre
2 FROM dan_p3.bien
3 JOIN dan_p3.vente USING (ID_BIEN)
4 JOIN dan_p3.commune USING (ID_COM)
5 JOIN dan_p3.region USING (ID_REG)
6 WHERE
7     DATE_VENTE BETWEEN '2020-01-01' AND '2020-06-30'
8     AND TYPE_LOCAL = 'appartement'
9 GROUP BY TYPE_LOCAL, REG
10 ORDER BY Nb_vente_1er_semestre DESC ;
```

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

	Type_de_local	Region	Nb_vente_1er_semestre
▶	Appartement	Île-de-France	13974
	Appartement	Provence-Alpes-Côte d'Azur	3639
	Appartement	Auvergne-Rhône-Alpes	3248
	Appartement	Nouvelle-Aquitaine	1931
	Appartement	Occitanie	1640
	Appartement	Pays de la Loire	1356
	Appartement	Hauts-de-France	1251
	Appartement	Grand Est	982
	Appartement	Bretagne	982
	Appartement	Normandie	860
	Appartement	Centre-Val de Loire	693
	Appartement	Bourgogne-Franche-Comté	376
	Appartement	Corse	222
	Appartement	Martinique	94
	Appartement	La Réunion	44
	Appartement	Guyane	34

Result 1 ×

Output

Action Output

#	Time	Action	Message
✓ 1	09:21:24	SELECT TYPE_LOCAL as Type_de_local, REG as Region, count(*) as Nb_vente_1er_semestre FRO...	16 row(s) returned



Parcours
Data Analyst
V2

OPENCLASSROOMS

4. Requête n°3 - Proportion des ventes d' appartements par le nombre de pièces

Nota : Nombre de pièces principales = 0 considéré comme une valeur aberrante et non retenue dans la préparation des données.

a. Jointure avec WHERE

Requete3 x

```
1 • SELECT b.TYPE_LOCAL as Type_de_local, b.TOT_PIECES as Nb_pièces,  
2       CONCAT(ROUND(count(*)*100/(select count(*)from dan_p3.vente),2),'%') as Pourcentage_vente  
3 FROM dan_p3.bien b, dan_p3.commune c, dan_p3.vente v  
4 WHERE  
5     v.DATE_VENTE BETWEEN '2020-01-01' AND '2020-06-30'  
6     AND b.ID_BIEN = v.ID_BIEN  
7     AND b.ID_COM = c.ID_COM  
8     AND b.TYPE_LOCAL = 'appartement'  
9 GROUP BY b.TYPE_LOCAL, b.TOT_PIECES  
10 ORDER BY b.TOT_PIECES ASC  
11 :
```

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

	Type_de_local	Nb_pièces	Pourcentage_vente
▶	Appartement	1	19.73%
	Appartement	2	28.65%
	Appartement	3	26.28%
	Appartement	4	13.07%
	Appartement	5	3.27%
	Appartement	6	0.60%
	Appartement	7	0.16%
	Appartement	8	0.05%
	Appartement	9	0.02%
	Appartement	10	0.01%
	Appartement	11	0.00%

Result 3 x

Output

Action Output

#	Time	Action	Message
✓ 6	12:03:10	SELECT b.TYPE_LOCAL as Type_de_local, b.TOT_PIECES as Nb_pièces, CONCAT(ROUND(c...	11 row(s) returned



Parcours
Data Analyst
V2

OPENCLASSROOMS

b. Jointure avec JOIN

Requête3b

```
1 • SELECT b.TYPE_LOCAL as Type_de_local, b.TOT_PIECES as Nb_pièces,  
2       CONCAT(ROUND(count(*)*100/(select count(*)from dan_p3.vente),2),'%') as Pourcentage_vente  
3 FROM dan_p3.bien b  
4 JOIN dan_p3.commune c USING (ID_COM)  
5 JOIN dan_p3.vente v USING (ID_BIEN)  
6 WHERE  
7       v.DATE_VENTE BETWEEN '2020-01-01' AND '2020-06-30'  
8       AND b.TYPE_LOCAL = 'appartement'  
9 GROUP BY b.TYPE_LOCAL, b.TOT_PIECES  
10 ORDER BY b.TOT_PIECES ASC  
11 ;
```

Result Grid

	Type_de_local	Nb_pièces	Pourcentage_vente
▶	Appartement	1	19.73%
	Appartement	2	28.65%
	Appartement	3	26.28%
	Appartement	4	13.07%
	Appartement	5	3.27%
	Appartement	6	0.60%
	Appartement	7	0.16%
	Appartement	8	0.05%
	Appartement	9	0.02%
	Appartement	10	0.01%
	Appartement	11	0.00%

Result 1

Output

Action Output

#	Time	Action	Message
✓ 1	12:11:19	SELECT b.TYPE_LOCAL as Type_de_local, b.TOT_PIECES as Nb_pièces, CONCAT(ROUND(cou...	11 row(s) returned



Parcours
Data Analyst
V2

OPENCLASSROOMS

5. Requête n°4 - Liste des 10 départements où le prix du mètre carré est le plus élevé

a. Jointure avec WHERE

Requete4a

```
1 • SELECT r.CODDEP as Code_departement, r.DEP as Departement,  
2         round(avg(v.VALEUR/b.SURF_CARREZ)) as Prix_m²_carrez,  
3         round(avg(v.VALEUR/b.SURF_RB)) as Prix_m²_reel_bati  
4 FROM dan_p3.bien b, dan_p3.commune c, dan_p3.vente v, dan_p3.region r  
5 WHERE  
6     v.DATE_VENTE BETWEEN '2020-01-01' AND '2020-06-30'  
7     AND b.ID_BIEN = v.ID_BIEN  
8     AND b.ID_COM = c.ID_COM  
9     AND c.ID_REG = r.ID_REG  
10 GROUP BY r.CODDEP, r.DEP  
11 ORDER BY Prix_m²_carrez DESC  
12 LIMIT 10  
13 ;
```

Result Grid

	Code_departement	Departement	Prix_m²_carrez	Prix_m²_reel_bati
▶	75	Paris	11716	11947
	92	Hauts-de-Seine	7219	7415
	94	Val-de-Marne	5343	5399
	6	Alpes-Maritimes	4701	4686
	74	Haute-Savoie	4667	4150
	93	Seine-Saint-Denis	4343	4369
	78	Yvelines	4221	4122
	69	Rhône	4059	4063
	2A	Corse-du-Sud	4027	3922
	33	Gironde	3764	3839

Result 1

Output

Action Output

#	Time	Action	Message
✓ 1	12:36:06	SELECT r.CODDEP as Code_departement, r.DEP as Departement, round(avg(v.VALEUR/b.SURF_C...	10 row(s) returned



Parcours
Data Analyst
V2

OPENCLASSROOMS

b. Jointure avec JOIN

Requete4b

```
1 • SELECT r.CODDEP as Code_departement, r.DEP as Departement,
2         round(avg(v.VALEUR/b.SURF_CARREZ)) as Prix_m²_carrez,
3         round(avg(v.VALEUR/b.SURF_RB)) as Prix_m²_reel_bati
4 FROM dan_p3.bien b
5 JOIN dan_p3.commune c USING (ID_COM)
6 JOIN dan_p3.vente v USING (ID_BIEN)
7 JOIN dan_p3.region r USING (ID_REG)
8 WHERE
9     v.DATE_VENTE BETWEEN '2020-01-01' AND '2020-06-30'
10 GROUP BY r.CODDEP, r.DEP
11 ORDER BY Prix_m²_carrez DESC
12 LIMIT 10
13 ;
```

Result Grid

	Code_departement	Departement	Prix_m²_carrez	Prix_m²_reel_bati
▶	75	Paris	11716	11947
	92	Hauts-de-Seine	7219	7415
	94	Val-de-Marne	5343	5399
	6	Alpes-Maritimes	4701	4686
	74	Haute-Savoie	4667	4150
	93	Seine-Saint-Denis	4343	4369
	78	Yvelines	4221	4122
	69	Rhône	4059	4063
	2A	Corse-du-Sud	4027	3922
	33	Gironde	3764	3839

Result 1

Output

Action Output

#	Time	Action	Message
✓ 1	14:27:12	SELECT r.CODDEP as Code_departement, r.DEP as Departement, round(avg(v.VALEUR/b.SURF_C...	10 row(s) returned



Parcours
Data Analyst
V2

OPENCLASSROOMS

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

a. WHERE et GROUP BY

Requete5a

```
1 • SELECT r.REG as Region, b.TYPE_LOCAL as Type_de_local,
2       round(avg(v.valeur/b.surf_carrez)) as Prix_m²_carrez,
3       round(avg(v.valeur/b.surf_RB)) as Prix_m²_reel_bati
4 FROM dan_p3.bien b, dan_p3.commune c, dan_p3.vente v, dan_p3.region r
5 WHERE
6     v.DATE_VENTE BETWEEN '2020-01-01' AND '2020-06-30'
7     AND b.ID_BIEN = v.ID_BIEN
8     AND b.ID_COM = c.ID_COM
9     AND c.ID_REG = r.ID_REG
10    AND b.TYPE_LOCAL='maison'
11    AND r.CODDEP IN ('75', '77', '78', '91', '92', '93', '94', '95')
12 GROUP BY r.REG, b.TYPE_LOCAL
13 ;
```

Result Grid

Region	Type_de_local	Prix_m²_carrez	Prix_m²_reel_bati
Île-de-France	Maison	3737	3999

Result 2

Output

Action Output

#	Time	Action	Message
✓ 1	14:33:01	SELECT r.REG as Region, b.TYPE_LOCAL as Type_de_local, round(avg(v.valeur/b.surf_carrez)) as ...	1 row(s) returned



Parcours
Data Analyst
V2

OPENCLASSROOMS

b. JOIN, WHERE et GROUP BY

Requete5b

```
1 • SELECT REG as Region, TYPE_LOCAL as Type_de_local,  
2         round(avg(VALEUR/SURF_CARREZ)) as Prix_m²_carrez,  
3         round(avg(VALEUR/SURF_RB)) as Prix_m²_reel_bati  
4 FROM bien  
5 JOIN vente USING (ID_BIEN)  
6 JOIN commune USING (ID_COM)  
7 JOIN region USING (ID_REG)  
8 WHERE  
9     DATE_VENTE BETWEEN '2020-01-01' AND '2020-06-30'  
10    AND TYPE_LOCAL='maison'  
11    AND CODDEP IN ('75', '77', '78', '91', '92', '93', '94', '95')  
12 GROUP BY REG, TYPE_LOCAL  
13 ;
```

Result Grid

	Region	Type_de_local	Prix_m²_carrez	Prix_m²_reel_bati
▶	Île-de-France	Maison	3737	3999

Result 1

Output

Action Output

#	Time	Action	Message
✓ 1	14:33:39	SELECT REG as Region, TYPE_LOCAL as Type_de_local, round(avg(VALEUR/SURF_CARREZ)) as...	1 row(s) returned



Parcours
Data Analyst
V2

OPENCLASSROOMS

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

Nota : Appartement ayant une valeur foncière 9millions pour une surface carrez de 10m² considéré comme valeur aberrante et supprimé.

a. WHERE et ORDER BY

Requete6a

```
1 • SELECT r.REG as Region, c.COM as Commune, b.TYPE_LOCAL as Type_de_local, b.SURF_CARREZ as Surface_carrez,
2       b.SURF_RB as Surface_reel_bati, v.VALEUR as Valeur_fonciere
3 FROM dan_p3.bien b, dan_p3.commune c, dan_p3.vente v, dan_p3.region r
4 WHERE
5       v.DATE_VENTE BETWEEN '2020-01-01' AND '2020-06-30'
6       AND b.ID_BIEN = v.ID_BIEN
7       AND b.ID_COM = c.ID_COM
8       AND c.ID_REG = r.ID_REG
9       AND b.TYPE_LOCAL = 'appartement'
10 ORDER BY v.VALEUR DESC
11 LIMIT 10
12 ;
```

Result Grid

	Region	Commune	Type_de_local	Surface_carrez	Surface_reel_bati	Valeur_fonciere
▶	Île-de-France	CORBEIL ESSONNES	Appartement	64	62	8600000
	Île-de-France	PARIS 07	Appartement	20.55	289	8577713
	Île-de-France	PARIS 17	Appartement	42.77	42	7620000
	Île-de-France	PARIS 06	Appartement	253.3	200	7600000
	Île-de-France	PARIS 01	Appartement	139.9	143	7535000
	Île-de-France	PARIS 16	Appartement	360.95	357	7420000
	Île-de-France	PARIS 16	Appartement	595	241	7200000
	Île-de-France	PARIS 01	Appartement	122.56	310	7050000
	Île-de-France	PARIS 01	Appartement	79.38	76	6600000
	Île-de-France	PARIS 16	Appartement	317.05	287	6500000

Result 2

Output

Action Output

#	Time	Action	Message
✓ 1	12:45:58	SELECT r.REG as Region, c.COM as Commune, b.TYPE_LOCAL as Type_de_local, b.SURF_CARRE...	10 row(s) returned



Parcours
Data Analyst
V2

OPENCLASSROOMS

b. JOIN et ORDER BY

Requete6b

```
1 • SELECT REG as Region, COM as Commune, TYPE_LOCAL as Type_de_local,  
2     SURF_CARREZ as Surface_carrez, SURF_RB as Surface_reel_bati, VALEUR as Valeur_fonciere  
3     FROM dan_p3.bien  
4     JOIN dan_p3.vente USING (ID_BIEN)  
5     JOIN dan_p3.commune USING (ID_COM)  
6     JOIN dan_p3.region USING (ID_REG)  
7     WHERE TYPE_LOCAL = 'appartement'  
8     ORDER BY VALEUR DESC  
9     LIMIT 10  
10    ;
```

Result Grid

	Region	Commune	Type_de_local	Surface_carrez	Surface_reel_bati	Valeur_fonciere
▶	Île-de-France	CORBEIL ESSONNES	Appartement	64	62	8600000
	Île-de-France	PARIS 07	Appartement	20.55	289	8577713
	Île-de-France	PARIS 17	Appartement	42.77	42	7620000
	Île-de-France	PARIS 06	Appartement	253.3	200	7600000
	Île-de-France	PARIS 01	Appartement	139.9	143	7535000
	Île-de-France	PARIS 16	Appartement	360.95	357	7420000
	Île-de-France	PARIS 16	Appartement	595	241	7200000
	Île-de-France	PARIS 01	Appartement	122.56	310	7050000
	Île-de-France	PARIS 01	Appartement	79.38	76	6600000
	Île-de-France	PARIS 16	Appartement	317.05	287	6500000

Result 1

Output

Action Output

#	Time	Action	Message
✓ 1	12:47:09	SELECT REG as Region, COM as Commune, TYPE_LOCAL as Type_de_local, SURF_CARREZ as S...	10 row(s) returned



Parcours
Data Analyst
V2

OPENCLASSROOMS

8. Requête n°7 - Taux d' évolution du nombre de ventes entre le premier et le second trimestre de 2020

a. CASE WHEN

```
1 • SELECT
2   concat(
3   round(
4   (count(CASE WHEN DATE_VENTE BETWEEN '2020-04-01' AND '2020-06-30' THEN 1 END)
5   - count(CASE WHEN DATE_VENTE BETWEEN '2020-01-01' and '2020-03-31' THEN 1 END))*100/
6   count(CASE WHEN DATE_VENTE BETWEEN '2020-01-01' AND '2020-03-31' THEN 1 END)
7   ,2)
8   , '%') AS Taux_evolution_ventes_entre_1er_et_2nd_trim
9 FROM dan_p3.vente
10 ;
```

Result Grid

Taux_evolution_ventes_entre_1er_et_2nd_trim
3.65%

Result 3 x

Output

Action Output

#	Time	Action	Message
✓ 1	16:15:56	SELECT concat(round((count(CASE WHEN DATE_VENTE BETWEEN '2020-04-01' AND '2020-06-30' ...	1 row(s) returned



Parcours
Data Analyst
V2

OPENCLASSROOMS

b. WITH

Requete7b

```
1 WITH ventes_1er_trim_table AS (  
2   SELECT count(*) as ventes_1er_trim  
3   FROM dan_p3.vente  
4   WHERE DATE_VENTE BETWEEN '2020-01-01' AND '2020-03-31'),  
5  
6   ventes_2nd_trim_table AS (  
7     SELECT count(*) as ventes_2nd_trim  
8     FROM dan_p3.vente  
9     WHERE DATE_VENTE BETWEEN '2020-04-01' AND '2020-06-30')  
10  
11 SELECT CONCAT(round((ventes_2nd_trim -ventes_1er_trim)*100/ventes_1er_trim,2),'%') AS Taux_evolution_ventes_entre_1er_et_2nd_trim  
12 FROM ventes_1er_trim_table, ventes_2nd_trim_table  
13 ;
```

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

Taux_evolution_ventes_entre_1er_et_2nd_trim
3.65%

Result 3

Output

Action Output

#	Time	Action	Message
1	14:49:01	WITH ventes_1er_trim_table AS (SELECT count(*) as ventes_1er_trim FROM dan_p3.vente WH...	1 row(s) returned



Parcours
Data Analyst
V2

OPENCLASSROOMS

9. Requête n°8 - Le classement des régions par rapport au prix au mètre carré des appartement de plus de 4 pièces

a. WHERE, GROUP BY

Requete8a*

```
1 SELECT r.REG as Region, b.TYPE_LOCAL as Type_de_local,
2       round(avg(v.valeur/b.surf_carrez)) as Prix_m²_carrez,
3       round(avg(v.valeur/b.surf_RB)) as Prix_m²_reel_bati
4 FROM dan_p3.bien b, dan_p3.commune c, dan_p3.vente v, dan_p3.region r
5 WHERE
6       v.DATE_VENTE BETWEEN '2020-01-01' AND '2020-06-30'
7       AND b.ID_BIEN = v.ID_BIEN
8       AND b.ID_COM = c.ID_COM
9       AND c.ID_REG = r.ID_REG
10      AND b.TYPE_LOCAL='appartement'
11      AND b.TOT_PIECES > 4
12 GROUP BY r.REG, b.TYPE_LOCAL
13 ORDER BY Prix_m²_carrez DESC
```

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

	Region	Type_de_local	Prix_m²_carrez	Prix_m²_reel_bati
▶	Île-de-France	Appartement	8770	8015
	La Réunion	Appartement	3642	3660
	Provence-Alpes-Côte d'Azur	Appartement	3588	3005
	Corse	Appartement	3105	3046
	Auvergne-Rhône-Alpes	Appartement	2891	2769
	Nouvelle-Aquitaine	Appartement	2465	2510
	Bretagne	Appartement	2412	2272
	Pays de la Loire	Appartement	2316	2187
	Hauts-de-France	Appartement	2190	2204
	Occitanie	Appartement	2097	2096
	Normandie	Appartement	2016	1994
	Grand Est	Appartement	1541	1313
	Centre-Val de Loire	Appartement	1453	1429
	Bourgogne-Franche-Comté	Appartement	1251	1069
	Martinique	Appartement	573	564

Result 3 x

Output

Action Output

#	Time	Action	Message
✓ 1	14:52:39	SELECT r.REG as Region, b.TYPE_LOCAL as Type_de_local, round(avg(v.valeur/b.surf_carrez)) as ...	15 row(s) returned



Parcours
Data Analyst
V2

OPENCLASSROOMS

b. JOIN, GROUP BY

Requête8b

```
1 SELECT REG as Region, TYPE_LOCAL as Type_de_local, round(avg(VALEUR/SURF_CARREZ)) as Prix_m²_carrez,  
2 round(avg(VALEUR/SURF_RB)) as Prix_m²_reel_bati  
3 FROM bien  
4 JOIN vente USING (ID_BIEN)  
5 JOIN commune USING (ID_COM)  
6 JOIN region USING (ID_REG)  
7 WHERE  
8 DATE_VENTE BETWEEN '2020-01-01' AND '2020-06-30'  
9 AND TYPE_LOCAL='appartement'  
10 AND TOT_PIECES > 4  
11 GROUP BY REG, TYPE_LOCAL  
12 ORDER BY Prix_m²_carrez DESC ;
```

Result Grid

	Region	Type_de_local	Prix_m²_carrez	Prix_m²_reel_bati
▶	Île-de-France	Appartement	8770	8015
	La Réunion	Appartement	3642	3660
	Provence-Alpes-Côte d'Azur	Appartement	3588	3005
	Corse	Appartement	3105	3046
	Auvergne-Rhône-Alpes	Appartement	2891	2769
	Nouvelle-Aquitaine	Appartement	2465	2510
	Bretagne	Appartement	2412	2272
	Pays de la Loire	Appartement	2316	2187
	Hauts-de-France	Appartement	2190	2204
	Occitanie	Appartement	2097	2096
	Normandie	Appartement	2016	1994
	Grand Est	Appartement	1541	1313
	Centre-Val de Loire	Appartement	1453	1429
	Bourgogne-Franche-Comté	Appartement	1251	1069
	Martinique	Appartement	573	564

Result 1

Output

Action Output

#	Time	Action	Message
✓ 1	14:53:23	SELECT REG as Region, TYPE_LOCAL as Type_de_local, round(avg(VALEUR/SURF_CARREZ)) as ...	15 row(s) returned



Parcours
Data Analyst
V2

OPENCLASSROOMS

10. Requête n°9 - Liste des communes ayant eu au moins 50 ventes au 1er trimestre

a. WHERE et HAVING

Requete9*

```
1 • SELECT c.CODPOS as Code_Postal, c.COM as Commune, count(*) AS nb_ventes_1er_trim
2 FROM dan_p3.bien b, dan_p3.vente v, dan_p3.commune c
3 WHERE
4     v.DATE_VENTE BETWEEN '2020-01-01' AND '2020-03-31'
5     AND v.ID_BIEN = b.ID_BIEN
6     AND b.ID_COM = c.ID_COM
7 GROUP BY c.CODPOS, c.COM
8 HAVING nb_ventes_1er_trim >=50
9 ORDER BY nb_ventes_1er_trim DESC
10 ;
```

Result Grid

Code_Postal	Commune	nb_ventes_1er_trim
75017	PARIS 17	228
75015	PARIS 15	215
75018	PARIS 18	209
75011	PARIS 11	169
75016	PARIS 16	165
75014	PARIS 14	146
75020	PARIS 20	127
75019	PARIS 19	116
75012	PARIS 12	110
75010	PARIS 10	108
75009	PARIS 09	106
92100	BOULOGNE-BILLANCOURT	97
75013	PARIS 13	94
75007	PARIS 07	87
75006	PARIS 06	86

Result 2

Output

Action Output

#	Time	Action	Message
✓ 1	16:30:53	SELECT c.CODPOS as Code_Postal, c.COM as Commune, count(*) AS nb_ventes_1er_trim FROM dan_p...	39 row(s) returned



Parcours
Data Analyst
V2

OPENCLASSROOMS

	Code_Postal	Commune	nb_ventes_1er_trim
▶	75017	PARIS 17	228
	75015	PARIS 15	215
	75018	PARIS 18	209
	75011	PARIS 11	169
	75016	PARIS 16	165
	75014	PARIS 14	146
	75020	PARIS 20	127
	75019	PARIS 19	116
	75012	PARIS 12	110
	75010	PARIS 10	108
	75009	PARIS 09	106
	92100	BOULOGNE-BILLANCOURT	97
	75013	PARIS 13	94
	75007	PARIS 07	87
	75006	PARIS 06	86
	92600	ASNIERES-SUR-SEINE	81
	13008	MARSEILLE 8EME	80
	92400	COURBEVOIE	80
	Code_Postal	Commune	nb_ventes_1er_trim
	75003	PARIS 03	79
	75005	PARIS 05	79
	6000	NICE	76
	38000	GRENOBLE	76
	33000	BORDEAUX	73
	13001	MARSEILLE 1ER	71
	13004	MARSEILLE 4EME	71
	92500	RUEIL-MALMAISON	68
	94300	VINCENNES	68
	13009	MARSEILLE 9EME	66
	93100	MONTREUIL	65
	34200	SETE	62
	75008	PARIS 08	62
	13600	LA CIOTAT	62
	75002	PARIS 02	61
	75004	PARIS 04	59
	92300	LEVALLOIS-PERRET	59
	6600	ANTIBES	53
	78000	VERSAILLES	53
	92800	PUTEAUX	53
	92130	ISSY-LES-MOULINEAUX	50



Parcours
Data Analyst
V2

OPENCLASSROOMS

b. JOIN et HAVING

Requete9b

```
1 • SELECT CODPOS as Code_Postal, COM as Commune, count(*) AS nb_ventes_1er_trim
2 FROM dan_p3.bien
3 JOIN dan_p3.vente USING (ID_BIEN)
4 JOIN dan_p3.commune USING (ID_COM)
5 WHERE
6     DATE_VENTE BETWEEN '2020-01-01' AND '2020-03-31'
7 GROUP BY CODPOS, COM
8 HAVING nb_ventes_1er_trim >=50
9 ORDER BY nb_ventes_1er_trim DESC
10 ;
```

Result Grid

	Code_Postal	Commune	nb_ventes_1er_trim
▶	75017	PARIS 17	228
	75015	PARIS 15	215
	75018	PARIS 18	209
	75011	PARIS 11	169
	75016	PARIS 16	165
	75014	PARIS 14	146
	75020	PARIS 20	127
	75019	PARIS 19	116
	75012	PARIS 12	110
	75010	PARIS 10	108
	75009	PARIS 09	106
	92100	BOULOGNE-BILLANCOURT	97
	75013	PARIS 13	94

Result 2

Output

Action Output

#	Time	Action	Message
✓ 1	17:49:14	SELECT CODPOS as Code_Postal, COM as Commune, count(*) AS nb_ventes_1er_trim FROM dan_p3.bien JOI...	39 row(s) returned



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

a. CREATE TEMPORARY TABLE

Requete10a

```
1 • CREATE TEMPORARY TABLE table_2p as (  
2     SELECT TOT_PIECES, round(avg(VALEUR/SURF_CARREZ)) as prix_m2_carrez_2  
3     FROM dan_p3.bien  
4     JOIN dan_p3.vente USING (ID_BIEN)  
5     WHERE  
6         DATE_VENTE BETWEEN '2020-01-01' AND '2020-06-30'  
7         AND TYPE_LOCAL = 'appartement'  
8         AND TOT_PIECES = 2  
9     GROUP BY TOT_PIECES)  
10 ;  
11 • CREATE TEMPORARY TABLE table_3p AS (  
12     SELECT TOT_PIECES, round(avg(VALEUR/SURF_CARREZ)) as prix_m2_carrez_3  
13     FROM dan_p3.bien  
14     JOIN dan_p3.vente USING (ID_BIEN)  
15     WHERE  
16         DATE_VENTE BETWEEN '2020-01-01' AND '2020-06-30'  
17         AND TYPE_LOCAL = 'appartement'  
18         AND TOT_PIECES = 3  
19     GROUP BY TOT_PIECES)  
20 ;  
21 • SELECT CONCAT(ROUND((prix_m2_carrez_3-prix_m2_carrez_2)*100/prix_m2_carrez_2),'%') as difference_prix_m2_2_3_pieces  
22 FROM table_2p, table_3p  
23 ;
```

Result Grid

difference_prix_m2_2_3_pieces
-12%

Result 1

Output

Action Output

#	Time	Action	Message
✓ 1	15:22:20	CREATE TEMPORARY TABLE table_2p as (SELECT TOT_PIECES, round(avg(VALEUR/SURF_CA...	1 row(s) affected Records: 1 Duplicates: 0 Warnings: 0
✓ 2	15:22:20	CREATE TEMPORARY TABLE table_3p AS (SELECT TOT_PIECES, round(avg(VALEUR/SURF_CA...	1 row(s) affected Records: 1 Duplicates: 0 Warnings: 0
✓ 3	15:22:20	SELECT CONCAT(ROUND((prix_m2_carrez_3-prix_m2_carrez_2)*100/prix_m2_carrez_2),'%') as differe...	1 row(s) returned



Parcours
Data Analyst
V2

OPENCLASSROOMS

b. WITH

Requete10b*

```
1 WITH table_2p AS (  
2     SELECT TOT_PIECES, round(avg(v.VALEUR/b.SURF_CARREZ)) as prix_m2_carrez_2  
3     FROM dan_p3.bien b, dan_p3.vente v  
4     WHERE  
5         v.DATE_VENTE BETWEEN '2020-01-01' AND '2020-06-30'  
6         AND v.ID_BIEN = b.ID_BIEN  
7         AND b.TYPE_LOCAL = 'appartement'  
8         AND b.TOT_PIECES = 2  
9     GROUP BY b.TOT_PIECES),  
10  
11 table_3p AS (  
12     SELECT TOT_PIECES, round(avg(v.VALEUR/b.SURF_CARREZ)) as prix_m2_carrez_3  
13     FROM dan_p3.bien b, dan_p3.vente v  
14     WHERE  
15         v.DATE_VENTE BETWEEN '2020-01-01' AND '2020-06-30'  
16         AND v.ID_BIEN = b.ID_BIEN  
17         AND b.TYPE_LOCAL = 'appartement'  
18         AND b.TOT_PIECES = 3  
19     GROUP BY b.TOT_PIECES)  
20  
21     SELECT CONCAT(ROUND((prix_m2_carrez_3-prix_m2_carrez_2)*100/prix_m2_carrez_2),'%') as difference_prix_m2_2_3_pieces  
22     FROM table_2p, table_3p;  
23
```

Result Grid

	difference_prix_m2_2_3_pieces
▶	-12%

Result 2

Output

Action Output

#	Time	Action	Message
✓ 1	16:02:33	WITH table_2p AS (SELECT TOT_PIECES, round(avg(v.VALEUR/b.SURF_CARREZ)) as prix_m2_c...	1 row(s) returned



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

a. WITH, RANK() OVER, WHERE

Requete11a*

```
1 WITH top3_tab as (  
2     SELECT r.CODDEP as Code_departement, r.DEP as Departement, c.COM as Commune,  
3           round(avg(v.VALEUR)) as Valeur_fondiere_moyenne,  
4           RANK() OVER (PARTITION BY r.CODDEP ORDER BY round(avg(v.VALEUR)) DESC) Classement  
5     FROM dan_p3.bien b, dan_p3.vente v, dan_p3.commune c, dan_p3.region r  
6     WHERE  
7         v.DATE_VENTE BETWEEN '2020-01-01' AND '2020-06-30'  
8         AND b.ID_BIEN = v.ID_BIEN  
9         AND b.ID_COM = c.ID_COM  
10        AND c.ID_REG = r.ID_REG  
11        AND r.CODDEP IN ('6', '13', '33', '59', '69')  
12    GROUP BY c.ID_COM  
13 )  
14 SELECT * FROM top3_tab WHERE Classement <4  
15 ;
```

Result Grid

	Code_departement	Departement	Commune	Valeur_fondiere_moyenne	Classement
▶	13	Bouches-du-Rhône	MARSEILLE 12EME	399500	1
	13	Bouches-du-Rhône	GIGNAC-LA-NERTHE	330000	2
	13	Bouches-du-Rhône	SAINT SAVOURNIN	314425	3
	33	Gironde	LA TESTE-DE-BUCH	705000	1
	33	Gironde	LEGE-CAP-FERRET	695051	2
	33	Gironde	VAYRES	335000	3
	59	Nord	BERSEE	433202	1
	59	Nord	CYSOING	408550	2
	59	Nord	HALLUIN	322250	3

Result 2

Output

Action Output

#	Time	Action	Message
✓ 1	10:25:40	WITH top3_tab as (SELECT r.CODDEP as Code_departement, r.DEP as Departement, c.COM as C...	15 row(s) returned



Parcours
Data Analyst
V2

OPENCLASSROOMS

Code_departement	Departement	Commune	Valeur_fonciere_moyenne	Classement
13	Bouches-du-Rhône	MARSEILLE 12EME	399500	1
13	Bouches-du-Rhône	GIGNAC-LA-NERTHE	330000	2
13	Bouches-du-Rhône	SAINT SAVOURNIN	314425	3
33	Gironde	LA TESTE-DE-BUCH	705000	1
33	Gironde	LEGE-CAP-FERRET	695051	2
33	Gironde	VAYRES	335000	3
59	Nord	BERSEE	433202	1
59	Nord	CYSOING	408550	2
59	Nord	HALLUIN	322250	3
6	Alpes-Maritimes	SAINT-JEAN-CAP-FERRAT	968750	1
6	Alpes-Maritimes	EZE	655000	2
6	Alpes-Maritimes	MOUANS-SARTOUX	476898	3
69	Rhône	VILLE SUR JARNIOUX	485300	1
69	Rhône	LYON 2EME	455217	2
69	Rhône	LYON 6EME	426968	3

b. WITH, RANK () OVER, JOIN, UNION

```
Requete11b
WITH table_dep6 as (
  SELECT CODDEP, DEP, COM, round(avg(VALEUR)) as Valeur_fonciere_moyenne, RANK() OVER (PARTITION BY CODDEP ORDER BY round(avg(VALEUR)) DESC) Classement
  FROM dan_p3.bien
  JOIN vente USING (ID_BIEN)
  JOIN commune USING (ID_COM)
  JOIN region USING (ID_REG)
  WHERE
    CODDEP = 6
  GROUP BY COM
),
table_dep13 as (
  SELECT CODDEP, DEP, COM, round(avg(VALEUR)) as Valeur_fonciere_moyenne, RANK() OVER (PARTITION BY CODDEP ORDER BY round(avg(VALEUR)) DESC) Classement
  FROM dan_p3.bien
  JOIN vente USING (ID_BIEN)
  JOIN commune USING (ID_COM)
  JOIN region USING (ID_REG)
  WHERE
    CODDEP = 13
  GROUP BY COM
),
```



Parcours
Data Analyst
V2

OPENCLASSROOMS

```
22
23 table_dep33 as (
24     SELECT CODEP, DEP, COM, round(avg(VALEUR)) as Valeur_fonciere_moyenne, RANK() OVER (PARTITION BY CODEP ORDER BY round(avg(VALEUR)) DESC) Classement
25     FROM dan_p3.bien
26     JOIN vente USING (ID_BIEN)
27     JOIN commune USING (ID_COM)
28     JOIN region USING (ID_REG)
29     WHERE
30         CODEP = 33
31     GROUP BY COM
32 ),
33
34 table_dep59 as (
35     SELECT CODEP, DEP, COM, round(avg(VALEUR)) as Valeur_fonciere_moyenne, RANK() OVER (PARTITION BY CODEP ORDER BY round(avg(VALEUR)) DESC) Classement
36     FROM dan_p3.bien
37     JOIN vente USING (ID_BIEN)
38     JOIN commune USING (ID_COM)
39     JOIN region USING (ID_REG)
40     WHERE
41         CODEP = 59
42     GROUP BY COM
43 ),
44
45 table_dep69 as (
46     SELECT CODEP, DEP, COM, round(avg(VALEUR)) as Valeur_fonciere_moyenne, RANK() OVER (PARTITION BY CODEP ORDER BY round(avg(VALEUR)) DESC) Classement
47     FROM dan_p3.bien
48     JOIN vente USING (ID_BIEN)
49     JOIN commune USING (ID_COM)
50     JOIN region USING (ID_REG)
51     WHERE
52         CODEP = 69
53     GROUP BY COM
54 )
55
56 SELECT CODEP as Code_Departement, DEP as Departement, Classement, COM as Commune, valeur_fonciere_moyenne as Valeur_fonciere_moyenne FROM table_dep6
57 WHERE Classement <4
58 UNION
59 SELECT CODEP as Code_Departement, DEP as Departement, Classement, COM as Commune, valeur_fonciere_moyenne as Valeur_fonciere_moyenne FROM table_dep13
60 WHERE Classement <4
61 UNION
62 SELECT CODEP as Code_Departement, DEP as Departement, Classement, COM as Commune, valeur_fonciere_moyenne as Valeur_fonciere_moyenne FROM table_dep33
63 WHERE Classement <4
64 UNION
65 SELECT CODEP as Code_Departement, DEP as Departement, Classement, COM as Commune, valeur_fonciere_moyenne as Valeur_fonciere_moyenne FROM table_dep59
66 WHERE Classement <4
67 UNION
68 SELECT CODEP as Code_Departement, DEP as Departement, Classement, COM as Commune, valeur_fonciere_moyenne as Valeur_fonciere_moyenne FROM table_dep69
69 WHERE Classement <4
70 ;
```



Parcours
Data Analyst
V2

OPENCLASSROOMS

Requete11b x

Limit to 1000 rows

```
1 TH table_dep6 as (  
2   SELECT CODDEP, DEP, COM, round(avg(VALEUR)) as Valeur_fonciere_moyenne, RANK() OVER (PARTITIO  
3   FROM dan_p3.bien  
4   JOIN vente USING (ID_BIEN)  
5   JOIN commune USING (ID_COM)  
6   JOIN region USING (ID_REG)
```

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

	Code_Departement	Departement	Classement	Commune	Valeur_fonciere_moyenne
▶	6	Alpes-Maritimes	1	SAINT-JEAN-CAP-FERRAT	968750
	6	Alpes-Maritimes	2	EZE	655000
	6	Alpes-Maritimes	3	MOUANS-SARTOUX	476898
	13	Bouches-du-Rhône	1	GIGNAC-LA-NERTHE	330000
	13	Bouches-du-Rhône	2	SAINT SAVOURNIN	314425
	13	Bouches-du-Rhône	3	CASSIS	313417
	33	Gironde	1	LEGE-CAP-FERRET	549501
	33	Gironde	2	VAYRES	335000
	33	Gironde	3	ARCACHON	307436
	59	Nord	1	BERSEE	433202
	59	Nord	2	CYSOING	408550
	59	Nord	3	HALLUIN	322250
	69	Rhône	1	VILLE SUR JARNIOUX	485300
	69	Rhône	2	LYON 2EME	455217
	69	Rhône	3	LYON 6EME	426968

Result 2 x

Output

Action Output

#	Time	Action	Message
✓ 1	10:51:16	WITH table_dep6 as (SELECT CODDEP, DEP, COM, round(avg(VALEUR)) as Valeur_fonciere_moye...	15 row(s) returned