

Requêtes SQL

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

```
SELECT (COUNT(id_disposition)) as "Nombre total appartements"  
FROM disposition JOIN bien ON disposition.id_local=bien.id_local  
WHERE type_local='Appartement';
```

The screenshot shows a PostgreSQL query editor interface. The top bar includes tabs for Dashboard, Properties, SQL, Statistics, Dependencies, and Dependents. The main query editor displays the following SQL query:

```
1 SELECT (COUNT(id_disposition)) as "Nombre total appartements"  
2 FROM disposition JOIN bien ON disposition.id_local=bien.id_local  
3 WHERE type_local='Appartement';
```

Below the query editor, the 'Data Output' tab is active, showing the results of the query. The results are displayed in a table with one row and one column:

Nombre total appartements
31362

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

```
CREATE TEMP TABLE "requete_2" AS  
SELECT (COUNT(id_disposition)) as "Nombre_total_appartements"  
FROM disposition JOIN bien ON disposition.id_local=bien.id_local  
WHERE type_local='Appartement';
```

```
SELECT ("nombre_pieces_principales"::integer) AS "Nombre de pièces principales",  
ROUND((COUNT (*)*100/AVG("Nombre_total_appartements") ::numeric), 3) as "Proportion  
(en %)"  
FROM "bien", "requete_2"  
WHERE type_local LIKE 'Appartement'  
GROUP BY nombre_pieces_principales  
ORDER BY nombre_pieces_principales ASC;
```

Dashboard Properties SQL Statistics Dependencies Dependents BDD/postgres@PostgreSQL 14 *

BDD/postgres@PostgreSQL 14 v

Query Editor Query History Scratch Pad

```

1 CREATE TEMP TABLE "requete_2" AS
2 SELECT (COUNT(id_disposition)) AS "Nombre_total_appartements"
3 FROM disposition JOIN bien ON disposition.id_local=bien.id_local
4 WHERE type_local='Appartement';
5
6
7 SELECT ("nombre_pieces_principales"::integer) AS "Nombre de pièces principales",
8 ROUND((COUNT(*)*100/AVG("Nombre_total_appartements")::numeric), 3) AS "Proportion (en %)"
9 FROM "bien", "requete_2"
10 WHERE type_local LIKE 'Appartement'
11 GROUP BY nombre_pieces_principales
12 ORDER BY nombre_pieces_principales ASC;

```

Data Output Explain Messages Notifications

	Nombre de pièces principales integer	Proportion (en %) numeric
1	0	0.096
2	1	21.459
3	10	0.006
4	11	0.003
5	2	31.114
6	3	28.525
7	4	14.208
8	5	3.552
9	6	0.650
10	7	0.172
11	8	0.054
12	9	0.026

✓ Successfully run. Total query runtime: 50 msec. 12 rows affected.

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

```

SELECT ((sum(valeur_fonciere)/sum(surface_reelle_batie))::money) AS "Prix_au_mètre_carré",
("code_departement") AS "Code département"
FROM disposition, bien, reference_cadastrale
WHERE disposition.id_local = bien.id_local AND disposition.id_reference_cadastrale =
reference_cadastrale.id_reference_cadastrale
GROUP BY "code_departement"
ORDER BY "Prix_au_mètre_carré" DESC
FETCH FIRST 10 ROWS ONLY;

```

Dashboard

Properties

SQL

Statistics

Dependencies

Dependents

BDD/postgres@PostgreSQL 14 *

</

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

```

SELECT "type_local", ROUND((AVG(valeur_fonciere::numeric/surface_reelle_batie)),2) as
prix_metre_carre
FROM reference_cadastrale, bien, disposition
WHERE disposition.id_local = bien.id_local AND disposition.id_reference_cadastrale =
reference_cadastrale.id_reference_cadastrale
AND "code_departement" IN ('75', '77', '78', '91', '92', '93', '94', '95')
AND type_local LIKE 'Maison'
GROUP BY type_local;

```

Dashboard Properties SQL Statistics Dependencies Dependents BDD/postgres@PostgreSQL 14 *

BDD/postgres@PostgreSQL 14

Query Editor Query History Scratch Pad

```

1 SELECT "type_local", ROUND((AVG(valeur_fonciere::numeric/surface_reelle_batie)),2) as prix_metre_carre
2 FROM reference_cadastrale, bien, disposition
3 WHERE disposition.id_local = bien.id_local AND disposition.id_reference_cadastrale = reference_cadastral
4 AND "code_departement" IN ('75', '77', '78', '91', '92', '93', '94', '95')
5 AND type_local LIKE 'Maison'
6 GROUP BY type_local;

```

Data Output Explain Messages Notifications

	type_local character varying	prix_metre_carre numeric
1	Maison	3997.71

5. Liste des 10 appartements les plus chers avec le département et le nombre de mètres carrés.

```

SELECT "type_local" as "Type de bien immobilier", "valeur_fonciere" as "Valeur foncière",
"code_departement" as "Code du département", "surface_reelle_batie" as
"Nombre_de_mètres_carrés"
FROM reference_cadastrale, bien, disposition
WHERE type_local LIKE 'Appartement'
AND disposition.id_local = bien.id_local AND disposition.id_reference_cadastrale =
reference_cadastrale.id_reference_cadastrale
ORDER BY valeur_fonciere DESC
FETCH FIRST 10 ROWS ONLY;

```

Dashboard Properties SQL Statistics Dependencies Dependents BDD/postgres@PostgreSQL 14 *

BDD/postgres@PostgreSQL 14 *

Query Editor Query History Scratch Pad

```

1 SELECT "type_local" as "Type de bien immobilier", "valeur_fonciere" as "Valeur foncière", "code_departement" as "Code du département", "nombre_m2" as "Nombre de mètres carrés"
2 FROM reference_cadastrale, bien, disposition
3 WHERE type_local LIKE 'Appartement'
4 AND disposition.id_local = bien.id_local AND disposition.id_reference_cadastrale = reference_cadastrale.id_reference_cadastrale
5 ORDER BY valeur_fonciere DESC
6 FETCH FIRST 10 ROWS ONLY;

```

Data Output Explain Messages Notifications

	Type de bien immobilier character varying	Valeur foncière money	Code du département character varying	Nombre de mètres carrés integer
1	Appartement	9 000 000,00 €	75	10
2	Appartement	8 600 000,00 €	91	62
3	Appartement	8 577 713,00 €	75	289
4	Appartement	7 620 000,00 €	75	42
5	Appartement	7 600 000,00 €	75	200
6	Appartement	7 535 000,00 €	75	143
7	Appartement	7 420 000,00 €	75	357
8	Appartement	7 200 000,00 €	75	241
9	Appartement	7 050 000,00 €	75	310
10	Appartement	6 600 000,00 €	75	76

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

```

CREATE TEMP TABLE requete_six_action_un AS
SELECT (COUNT(*)) as Vd
FROM "disposition"
WHERE "date_mutation" LIKE '2020-01%' OR "date_mutation" LIKE '2020-02%' OR
"date_mutation" LIKE '2020-03%';

```

```

CREATE TEMP TABLE requete_six_action_deux AS
SELECT (COUNT(*)) as Va
FROM "disposition"
WHERE "date_mutation" LIKE '2020-04%' OR "date_mutation" LIKE '2020-05%' OR
"date_mutation" LIKE '2020-06%';

```

```

SELECT ROUND(((requete_six_action_deux.Va -
requete_six_action_un.Vd)*100/requete_six_action_un.Vd ::decimal), 2) as "Taux évolution entre
le premier et le deuxième trimestre"
FROM requete_six_action_un, requete_six_action_deux

```

Dashboard Properties SQL Statistics Dependencies Dependents BDD/postgres@PostgreSQL 14 *

BDD/postgres@PostgreSQL 14

Query Editor Query History Scratch Pad

```

1 CREATE TEMP TABLE requete_six_action_un AS
2 SELECT (COUNT(*)) as Vd
3 FROM "disposition"
4 WHERE "date_mutation" LIKE '2020-01%' OR "date_mutation" LIKE '2020-02%' OR
5 "date_mutation" LIKE '2020-03%';
6
7 CREATE TEMP TABLE requete_six_action_deux AS
8 SELECT (COUNT(*)) as Va
9 FROM "disposition"
10 WHERE "date_mutation" LIKE '2020-04%' OR "date_mutation" LIKE '2020-05%' OR
11 "date_mutation" LIKE '2020-06%';
12
13 SELECT ROUND(((requete_six_action_deux.Va -
14 requete_six_action_un.Vd)*100/requete_six_action_un.Vd ::decimal), 2) as "Taux évolution entre le premier
15 FROM requete_six_action_un, requete_six_action_deux

```

Data Output Explain Messages Notifications

Taux évolution entre le premier et le deuxième trimestre numeric	
1	3.66

PostgreSQL 14/BDD - Database connected.

7. Liste des communes où le nombre de ventes a augmenté d'au moins 20% entre le premier et le second trimestre de 2020.

```

CREATE TEMP TABLE requete_sept_action_un AS
SELECT "commune", (COUNT(*)) as Vd
FROM "disposition" JOIN "adresse" ON disposition.id_adresse=adresse.id_adresse
WHERE "date_mutation" LIKE '2020-01%' OR "date_mutation" LIKE '2020-02%' OR
"date_mutation" LIKE '2020-03%'
GROUP BY "commune";

```

```

CREATE TEMP TABLE requete_sept_action_deux AS
SELECT "commune", (COUNT(*)) as Va
FROM "disposition" JOIN "adresse" ON disposition.id_adresse=adresse.id_adresse
WHERE "date_mutation" LIKE '2020-04%' OR "date_mutation" LIKE '2020-05%' OR
"date_mutation" LIKE '2020-06%'
GROUP BY "commune";

```

```

SELECT requete_sept_action_un.commune, ROUND(((requete_sept_action_deux.Va-
requete_sept_action_un.Vd)*100/requete_sept_action_un.Vd ::decimal),2) as "Taux évolution"
FROM requete_sept_action_un, requete_sept_action_deux
WHERE requete_sept_action_un.commune=requete_sept_action_deux.commune
GROUP BY requete_sept_action_un.commune, ((requete_sept_action_deux.Va-
requete_sept_action_un.Vd)*100/requete_sept_action_un.Vd ::decimal)
HAVING ((requete_sept_action_deux.Va-
requete_sept_action_un.Vd)*100/requete_sept_action_un.Vd ::decimal)>20
ORDER BY commune ASC;

```

Dashboard Properties SQL Statistics Dependencies Dependents BDD/postgres@PostgreSQL 14 *

BDD/postgres@PostgreSQL 14

Query Editor Query History Scratch Pad

```

1 CREATE TEMP TABLE requete_sept_action_un AS
2 SELECT "commune", (COUNT (*)) as Vd
3 FROM "disposition" JOIN "adresse" ON disposition.id_adresse=adresse.id_adresse
4 WHERE "date_mutation" LIKE '2020-01%' OR "date_mutation" LIKE '2020-02%' OR "date_mutation" LIKE '2020-
5 GROUP BY "commune";
6
7 CREATE TEMP TABLE requete_sept_action_deux AS
8 SELECT "commune", (COUNT (*)) as Va
9 FROM "disposition" JOIN "adresse" ON disposition.id_adresse=adresse.id_adresse
10 WHERE "date_mutation" LIKE '2020-04%' OR "date_mutation" LIKE '2020-05%' OR "date_mutation" LIKE '2020-
11 GROUP BY "commune";
12
13 SELECT requete_sept_action_un.commune, ROUND(((requete_sept_action_deux.Va-requete_sept_action_un.Vd)*1
14 FROM requete_sept_action_un, requete_sept_action_deux
15 WHERE requete_sept_action_un.commune=requete_sept_action_deux.commune
16 GROUP BY requete_sept_action_un.commune, ((requete_sept_action_deux.Va-requete_sept_action_un.Vd)*100/r
17 HAVING ((requete_sept_action_deux.Va-requete_sept_action_un.Vd)*100/requete_sept_action_un.Vd)::decimal
18 ORDER BY commune ASC;

```

Data Output Explain Messages Notifications

	commune character varying	Taux évolution numeric
1	ABBEVILLE	350.00
2	ABLON-SUR-SEINE	160.00
3	AGDE	104.55
4	AIGUES-MORTES	83.33
5	AIRE-SUR-LADOUR	400.00
6	ALENCON	100.00

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

```

CREATE TEMP TABLE requete_huit_action_un AS
SELECT "nombre_pieces_principales", AVG(valeur_fonciere::numeric
/surface_reelle_batie::numeric) as prix_2
FROM "disposition" JOIN "bien" ON disposition.id_local=bien.id_local
WHERE type_local LIKE 'Appartement'
AND "nombre_pieces_principales" LIKE '2'
GROUP BY "nombre_pieces_principales";

```

```

CREATE TEMP TABLE requete_huit_action_deux AS
SELECT "nombre_pieces_principales",
AVG(valeur_fonciere::numeric/surface_reelle_batie::numeric) as prix_3
FROM "disposition" JOIN "bien" ON disposition.id_local=bien.id_local
WHERE type_local LIKE 'Appartement'
AND "nombre_pieces_principales" LIKE '3'
GROUP BY "nombre_pieces_principales";

```

```

SELECT ROUND((prix_3-prix_2)*100/( prix_2),2) as difference_pourcentage
FROM requete_huit_action_un, requete_huit_action_deux;

```

Dashboard Properties SQL Statistics Dependencies Dependents BDD/postgres@PostgreSQL 14 *

BDD/postgres@PostgreSQL 14

Query Editor Query History Scratch Pad

```

1 CREATE TEMP TABLE requete_huit_action_un AS
2 SELECT "nombre_pieces_principales", AVG(valeur_fonciere::numeric)
3 /surface_reelle_batie ::numeric) as prix_2
4 FROM "disposition" JOIN "bien" ON disposition.id_local=bien.id_local
5 WHERE type_local LIKE 'Appartement'
6 AND "nombre_pieces_principales" LIKE '2'
7 GROUP BY "nombre_pieces_principales";
8
9 CREATE TEMP TABLE requete_huit_action_deux AS
10 SELECT "nombre_pieces_principales",
11 AVG(valeur_fonciere::numeric/surface_reelle_batie ::numeric) as prix_3
12 FROM "disposition" JOIN "bien" ON disposition.id_local=bien.id_local
13 WHERE type_local LIKE 'Appartement'
14 AND "nombre_pieces_principales" LIKE '3'
15 GROUP BY "nombre_pieces_principales";
16
17 SELECT ROUND((prix_3-prix_2)*100/(prix_2),2) as difference_pourcentage
18 FROM requete_huit_action_un, requete_huit_action_deux;

```

Data Output Explain Messages Notifications

	difference_pourcentage
1	-13.13

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

```

CREATE TEMP TABLE requete_neuf_action_un AS
SELECT "commune", "code_departement", "valeur_fonciere"
FROM "disposition"
JOIN "reference_cadastrale" ON disposition.id_reference_cadastrale=
reference_cadastrale.id_reference_cadastrale
JOIN "adresse" ON disposition.id_adresse= adresse.id_adresse
WHERE "code_departement" IN ('6', '13', '33', '59', '69');

```

```

CREATE TEMP TABLE requete_neuf_action_six AS
SELECT "commune", AVG("valeur_fonciere":: numeric)
FROM requete_neuf_action_un
GROUP BY "commune";

```

```

CREATE TEMP TABLE requete_neuf_action_sept AS
SELECT DISTINCT requete_neuf_action_un.commune, "code_departement", "avg"
FROM requete_neuf_action_un, requete_neuf_action_six
WHERE requete_neuf_action_un.commune=requete_neuf_action_six.commune
ORDER BY "code_departement";

```

```

CREATE TEMP TABLE requete_neuf_action_huit AS
SELECT "commune", "code_departement", ROUND("avg",2) as "Moyenne des valeurs foncières
par commune", rank() OVER (PARTITION BY "code_departement" ORDER BY "avg" DESC) as
"Classement par commune"
FROM requete_neuf_action_sept;

```


SELECT *
 FROM requete_neuf_action_huit
 WHERE "Classement par commune"='1'OR "Classement par commune"='2'OR "Classement par commune"='3';

Dashboard Properties SQL Statistics Dependencies Dependents BDD/postgres@PostgreSQL 14 *

BDD/postgres@PostgreSQL 14

Query Editor Query History Scratch Pad

```

1 CREATE TEMP TABLE requete_neuf_action_un AS
2 SELECT "commune", "code_departement", "valeur_fonciere"
3 FROM "disposition"
4 JOIN "reference_cadastrale" ON disposition.id_reference_cadastrale= reference_cadastrale.id_reference_c
5 JOIN "adresse" ON disposition.id_adresse= adresse.id_adresse
6 WHERE "code_departement" IN ('6', '13', '33', '59', '69');
7
8 CREATE TEMP TABLE requete_neuf_action_six AS
9 SELECT "commune", AVG("valeur_fonciere":: numeric)
10 FROM requete_neuf_action_un
11 GROUP BY "commune";
12
13 CREATE TEMP TABLE requete_neuf_action_sept AS
14 SELECT DISTINCT requete_neuf_action_un.commune, "code_departement", "avg"
15

```

Data Output Explain Messages Notifications

	commune character varying	code_departement character varying	Moyenne des valeurs foncières par commune numeric	Classement par commune bigint	
1	GIGNAC-LA-NERTHE	13	330000.00	1	
2	SAINT SAVOURNIN	13	314425.00	2	
3	CASSIS	13	313416.88	3	
4	LEGE-CAP-FERRET	33	549500.64	1	
5	VAYRES	33	335000.00	2	
6	ARCACHON	33	307435.93	3	
7	BERSEE	59	433202.00	1	
8	CYSOING	59	408550.00	2	
9	HALLUIN	59	322250.00	3	
10	SAINT-JEAN-CAP-FERRAT	6	968750.00	1	

Help

Dashboard Properties SQL Statistics Dependencies Dependents BDD/postgres@PostgreSQL 14 *

BDD/postgres@PostgreSQL 14

Query Editor Query History Scratch Pad

```

12
13 CREATE TEMP TABLE requete_neuf_action_sept AS
14 SELECT DISTINCT requete_neuf_action_un.commune, "code_departement", "avg"
15 FROM requete_neuf_action_un, requete_neuf_action_six
16 WHERE requete_neuf_action_un.commune=requete_neuf_action_six.commune
17 ORDER BY "code_departement";
18
19 CREATE TEMP TABLE requete_neuf_action_huit AS
20 SELECT "commune", "code_departement", ROUND("avg",2) as "Moyenne des valeurs foncières par commune", ra
21 FROM requete_neuf_action_sept;
22
23 SELECT *
24 FROM requete_neuf_action_huit
25 WHERE "Classement par commune"='1'OR "Classement par commune"='2'OR "Classement par commune"='3';
26

```

Data Output Explain Messages Notifications

	commune character varying	code_departement character varying	Moyenne des valeurs foncières par commune numeric	Classement par commune bigint	
7	BERSEE	59	433202.00	1	
8	CYSOING	59	408550.00	2	
9	HALLUIN	59	322250.00	3	
10	SAINT-JEAN-CAP-FERRAT	6	968750.00	1	
11	EZE	6	655000.00	2	
12	MOUANS-SARTOUX	6	476898.10	3	
13	VILLE SUR JARNIOUX	69	485300.00	1	
14	LYON 2EME	69	455217.27	2	
15	LYON 6EME	69	426968.25	3	