

Requêtes SQL

Requête 1 :

Lister les numéros de contrats (contrat_ID) avec leur surface pour la commune de Caen.

Réponse

```
SELECT c.contrat_id, c.surface, r.com_nom_maj_court  
FROM contrat AS c  
JOIN region AS r  
ON c.code_dep_code_commune = r.code_dep_code_commune  
WHERE r.com_nom_maj_court = 'CAEN';
```

Résultat : 4 enregistrements

The screenshot shows the pgAdmin 4 interface with a query editor and a data output viewer.

Query Editor:

```
gestion_logement/postgres@PostgreSQL 18
SELECT c.contrat_id, c.surface, r.com_nom_maj_court
FROM contrat AS c
JOIN region AS r
ON c.code_dep_code_commune = r.code_dep_code_commune
WHERE r.com_nom_maj_court = 'CAEN';
```

Data Output:

	contrat_id	surface	com_nom_maj_court
1	103791	35	CAEN
2	103792	99	CAEN
3	103793	40	CAEN
4	103794	20	CAEN

Total rows: 4 Query complete 00:00:00.071 CRLF Ln 6, Col 1

Requête 2 :

Lister les numéros de contrats avec le type de contrat et leur formule pour les maisons du département 71.

Réponse

```
SELECT
```

```
    c.contrat_id AS "Numéro de contrat",
    c.type_contrat AS "Type de contrat",
    c.formule AS "Formule",
    r.dep_code AS "Département"

FROM contrat AS c

INNER JOIN region AS r

    ON c.code_dep_code_commune = r.code_dep_code_commune

WHERE c.type_local = 'Maison'

AND r.dep_code = '71';
```

Résultat : 4 enregistrements

The screenshot shows the pgAdmin 4 interface with a query editor and a data output viewer.

Query Editor:

```
gestion_logement/postgres@PostgreSQL 18
SELECT
    c.contrat_id AS "Numéro de contrat",
    c.type_contrat AS "Type de contrat",
    c.formule AS "Formule",
    r.dep_code AS "Département"
FROM contrat AS c
INNER JOIN region AS r
    ON c.code_dep_code_commune = r.code_dep_code_commune
WHERE c.type_local = 'Maison'
AND r.dep_code = '71';
1
2
3
4
5
6
7
8
9
10
11
12
```

Data Output:

	Numéro de contrat integer	Type de contrat character varying (30)	Formule character varying (30)	Département character varying (3)
1	114768	Residence principale	Integral	71
2	114779	Residence principale	Classique	71
3	114782	Residence principale	Classique	71
4	114812	Residence principale	Integral	71

Total rows: 4 Query complete 00:00:00.058 CRLF Ln 10, Col 25

Requête 3 :

Lister le nom des régions de France.

Réponse

```
SELECT DISTINCT reg_nom  
FROM region;
```

☞ distinct : évite les doublons

Résultat : 19 enregistrements

The screenshot shows a PostgreSQL database client interface. The top bar displays the connection information: 'gestion_logement/postgres@PostgreSQL 18'. Below the toolbar, the 'Query' tab is selected, showing the executed SQL code:

```
1 SELECT DISTINCT reg_nom  
2 FROM region;  
3  
4
```

The main area displays the results of the query. A table titled 'reg_nom' (character varying (50)) contains 19 rows of data, numbered 1 to 19. The data includes various French regions and collectivities:

	reg_nom
1	Hauts-de-France
2	La Réunion
3	Bretagne
4	Mayotte
5	Auvergne-Rhône-Alpes
6	Ile-de-France
7	Grand Est
8	Corse
9	Collectivités d'outre-mer
10	Guyane
11	Bourgogne-Franche-Comté
12	Provence-Alpes-Côte d'A...
13	
14	
15	
16	
17	
18	
19	

At the bottom of the results pane, it says 'Showing rows: 1 to 19' and 'Page No: 1 of 1'. The status bar at the bottom indicates 'Total rows: 19' and 'Query complete 00:00:00.133'.

Requête 4 :

Quels sont les 5 contrats qui ont les surfaces les plus élevées ?

Réponse

```
SELECT *  
FROM contrat  
ORDER BY surface DESC  
LIMIT 5;
```

Résultat

The screenshot shows the pgAdmin interface with a query editor and a results grid. The query is:

```
1 SELECT *
2 FROM contrat
3 ORDER BY surface DESC
4 LIMIT 5;
```

The results grid displays 5 rows of data from the 'contrat' table:

	contrat_id [PK] integer	no_voie integer	b_t_q character (1)	type_de_voie character varying	voie character varying (15)	code_dep_code_commu character varying (15)	code_postal character varying (5)	surface integer	type_local character varying (30)
1	104211	34	B	CRS	VICTOR HUGO	17421	17600	815	Appartement
2	105463	5229	[null]	RUE	DE LA PRIVADIÈRE	30126	30190	742	Maison
3	130878	23	[null]	BD	DE BEAUSEJOUR	75116	75016	595	Appartement
4	100822	5	A	RUE	FERNAND BAZIN	2201	2600	570	Maison
5	109872	18	[null]	RUE	PAUL BELLAMY	44109	44000	559	Appartement

Requête 5 :

Quel est le prix moyen de la cotisation mensuelle ?

Réponse

```
SELECT
    ROUND(AVG(prix_cotisation_mensuel), 2) AS "Moyenne de cotisation mensuelle"
    COUNT(*) AS "Nombre de contrats"
FROM contrat;
```

Résultat

The screenshot shows the pgAdmin interface with a query editor and a results grid. The query is:

```
1 SELECT
2     ROUND(AVG(prix_cotisation_mensuel), 2) AS "Moyenne de cotisation mensuelle",
3     COUNT(*) AS "Nombre de contrats"
4     FROM contrat;
```

The results grid displays 1 row of data:

	Moyenne de cotisation mensuelle numeric	Nombre de contrats bigint
1	19.33	30326

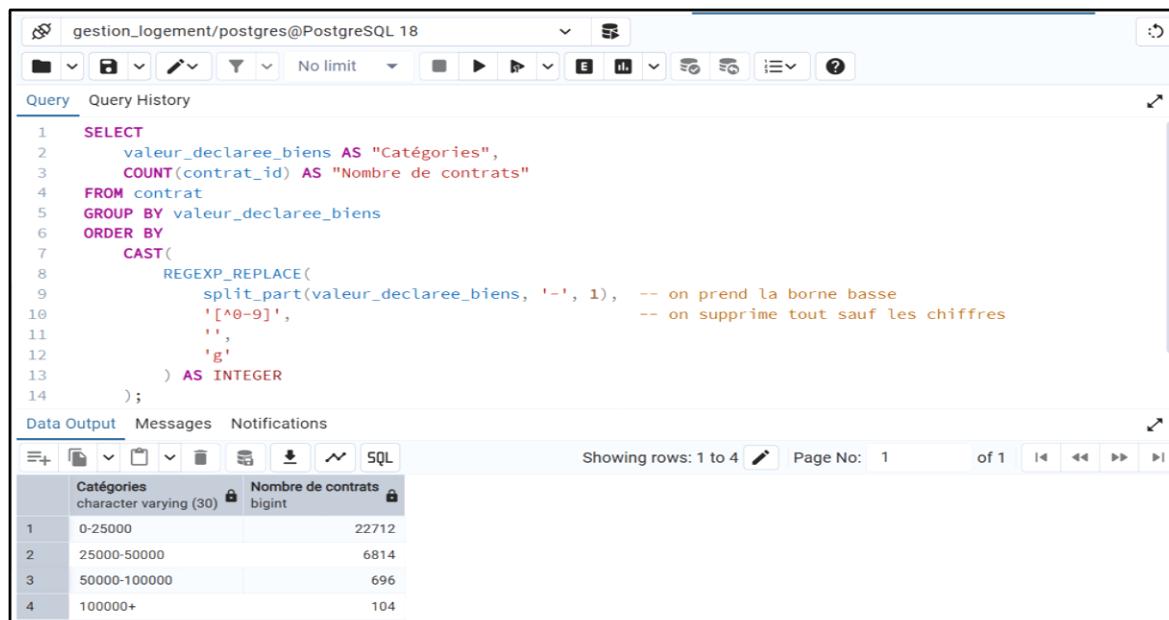
Requête 6 :

Quel est le nombre de contrats pour chaque catégorie de prix de la valeur déclarée des biens ?

Réponse

```
SELECT
    valeur_declaree_biens AS "Catégories",
    COUNT(contrat_id) AS "Nombre de contrats"
FROM contrat
GROUP BY valeur_declaree_biens
ORDER BY
    CAST(
        REGEXP_REPLACE(
            split_part(valeur_declaree_biens, '-', 1), '[^0-9]', '', 'g' ) AS INTEGER );
```

Résultat



The screenshot shows the pgAdmin 4 interface with a query editor and a results table.

Query Editor:

```
1 SELECT
2     valeur_declaree_biens AS "Catégories",
3     COUNT(contrat_id) AS "Nombre de contrats"
4 FROM contrat
5 GROUP BY valeur_declaree_biens
6 ORDER BY
7     CAST(
8         REGEXP_REPLACE(
9             split_part(valeur_declaree_biens, '-', 1), -- on prend la borne basse
10            '[^0-9]', -- on supprime tout sauf les chiffres
11            '',
12            'g'
13        ) AS INTEGER
14    );
```

Results Table:

	Catégories	Nombre de contrats
1	0-25000	22712
2	25000-50000	6814
3	50000-100000	696
4	100000+	104

Explications :

- **split_part(valeur_declaree_biens, '-', 1)** → prend la **partie avant le tiret** ("0" dans "0-25000", "25000" dans "25000-50000", et "100000+" reste "100000+").
- **REGEXP_REPLACE(..., '[^0-9]', '', 'g')** → remplace **tout ce qui n'est pas un chiffre** ('[^0-9]') par **rien** (""), partout dans la chaîne ('g' pour "global")
- **CAST(... AS INTEGER)** → convertit le texte nettoyé en **nombre entier**.

- **ORDER BY** → trie du plus petit au plus grand (tu peux ajouter DESC à la fin si tu veux l'inverse).

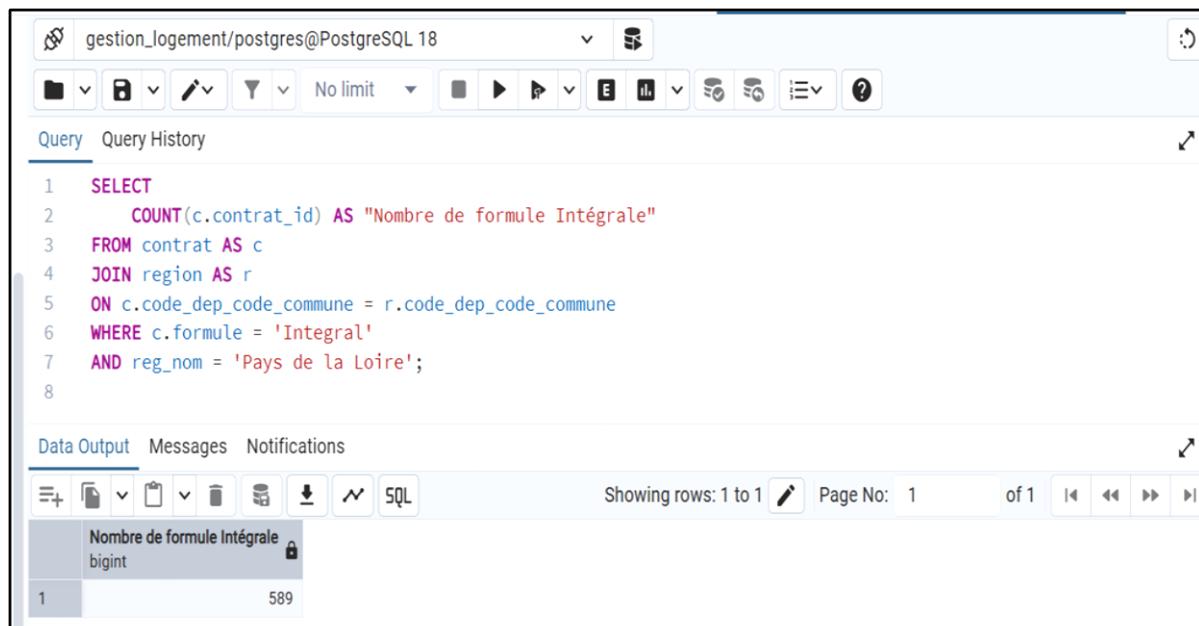
Requête 7 :

Quel est le nombre de formules “Integral” sur la région Pays de la Loire ?

Réponse

```
SELECT
    COUNT(c.contrat_id) AS "Nombre de formule Intégrale"
FROM contrat AS c
JOIN region AS r
ON c.code_dep_code_commune = r.code_dep_code_commune
WHERE c.formule = 'Integral'
AND reg_nom = 'Pays de la Loire';
```

Résultat



The screenshot shows a PostgreSQL database interface with the following details:

- Connection:** gestion_logement/postgres@PostgreSQL 18
- Toolbar:** Includes icons for file operations, search, and various database functions.
- Query Tab:** Contains the executed SQL query.
- Query History:** Shows the history of previous queries.
- Data Output:** Displays the result of the query in a table format.
- Messages:** Shows notifications or errors.
- Notifications:** Shows system notifications.
- Bottom Bar:** Includes icons for file operations, a SQL button, and navigation controls.
- Result Table:**

	Nombre de formule Intégrale
1	589

Requête 8 :

Lister les numéros de contrats avec le type de contrat et leur formule pour les maisons du département 71.

Réponse

```
SELECT
    c.contrat_id AS "Numéro de contrat",
    c.type_contrat AS "Type de contrat",
    c.formule AS "Formule",
    r.dep_code AS "Code département"

FROM contrat AS c

JOIN region AS r

ON c.code_dep_code_commune = r.code_dep_code_commune

WHERE r.dep_code = '71';

AND c.type_local = 'Maison';
```

Résultat

The screenshot shows a PostgreSQL client interface with the following details:

- Query Bar:** Displays the executed SQL query.
- Data Output:** Shows the results of the query, which are four rows of data from the 'contrat' table.
- Table Headers:** Numéro de contrat, Type de contrat, Formule, Code département.
- Table Data:**

	Numéro de contrat integer	Type de contrat character varying (30)	Formule character varying (30)	Code département character varying (3)
1	114768	Residence principale	Integral	71
2	114779	Residence principale	Classique	71
3	114782	Residence principale	Classique	71
4	114812	Residence principale	Integral	71

Requête 9 :

Quelle est la surface moyenne des contrats à Paris ?

Réponse

SELECT

ROUND(AVG(c.surface), 2) **AS** "Surface moyenne des contrats de Paris"

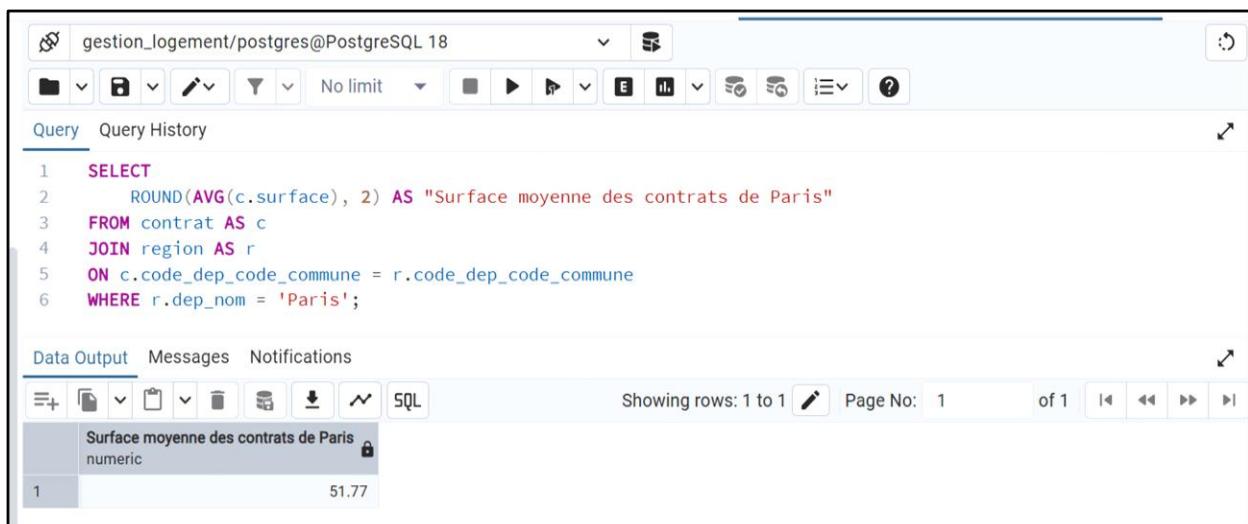
FROM contrat **AS** c

JOIN region **AS** r

ON c.code_dep_code_commune = r.code_dep_code_commune

WHERE r.dep_nom = 'Paris';

Résultat



The screenshot shows a PostgreSQL SQL editor interface. The query window contains the following SQL code:

```
1 SELECT
2     ROUND(AVG(c.surface), 2) AS "Surface moyenne des contrats de Paris"
3 FROM contrat AS c
4 JOIN region AS r
5 ON c.code_dep_code_commune = r.code_dep_code_commune
6 WHERE r.dep_nom = 'Paris';
```

The results pane shows a single row of data:

	Surface moyenne des contrats de Paris
1	51.77

Below the results, the status bar indicates "Showing rows: 1 to 1" and "Page No: 1 of 1".

Requête 10 :

Classement des 10 départements où le prix moyen de la cotisation est le plus élevé.

Réponse

SELECT

ROUND(AVG(prix_cotisation_mensuel),2) **AS** "Cotisation mensuelle moyenne",
r.dep_nom **AS** "Département"

FROM contrat **AS** c

JOIN region **AS** r

ON c.code_dep_code_commune = r.code_dep_code_commune

GROUP BY

r.dep_nom

ORDER BY "Cotisation mensuelle moyenne" **DESC**

LIMIT 10;

Résultat

The screenshot shows a PostgreSQL database interface with the following details:

- Connection:** gestion_logement/postgres@PostgreSQL 18
- Toolbar:** Includes icons for file operations, search, and various database functions.
- Query History:** Shows the history of queries run.
- Query Editor:** Displays the SQL code:

```
1 SELECT
2     ROUND(AVG(prix_cotisation_mensuel),2) AS "Cotisation mensuelle moyenne",
3     r.dep_nom AS "Département"
4 FROM contrat AS c
5 JOIN region AS r
6 ON c.code_dep_code_commune = r.code_dep_code_commune
7 GROUP BY
8     r.dep_nom
9 ORDER BY "Cotisation mensuelle moyenne" DESC
10 LIMIT 10;
11 |
```
- Data Output:** Shows the results of the query in a table format:

	Cotisation mensuelle moyenne numeric	Département character varying (50)
1	36.40	Paris
2	26.27	Hauts-de-Seine
3	19.82	Val-de-Marne
4	18.89	Yvelines
5	18.49	Rhône
6	18.24	Ain
7	18.14	Alpes-Maritimes
8	17.99	Charente-Maritime
- Information:** Total rows: 10, Query complete 00:00:00.130, CRLF, Ln 11, Col 1.

Requête 11 :

Liste des communes ayant eu au moins 150 contrats.

Réponse

```
SELECT
    r.com_nom_maj_court AS "Commune",
    COUNT(c.contrat_id) AS "Nombre de contrats"
FROM contrat AS c
JOIN region AS r
ON c.code_dep_code_commune = r.code_dep_code_commune
GROUP BY "Commune"
HAVING COUNT(c.contrat_id) >= 150;
```

Résultat

The screenshot shows a PostgreSQL database interface in pgAdmin. The title bar indicates the connection is to 'gestion_logement/postgres@PostgreSQL 18'. The toolbar has various icons for file operations, search, and navigation. Below the toolbar, there are tabs for 'Query' (which is selected), 'Query History', 'Data Output', 'Messages', and 'Notifications'. The main area displays a SQL query and its results.

```
1 SELECT
2     r.com_nom_maj_court AS "Commune",
3     COUNT(c.contrat_id) AS "Nombre de contrats"
4 FROM contrat AS c
5 JOIN region AS r
6 ON c.code_dep_code_commune = r.code_dep_code_commune
7 GROUP BY "Commune"
8 HAVING COUNT(c.contrat_id) >= 150;
```

	Commune	Nombre de contrats
1	PARIS 15	407
2	LILLE	161
3	COURBEVOIE	163
4	NICE	387
5	PARIS 16	394
6	PARIS 3	159
7	NANTES	291
8	TOULON	170
9	PARIS 14	222

Total rows: 20 Query complete 00:00:00.134 CRLF Ln 10, Col 1

Requête 12 :

Quel est le nombre de contrats pour chaque région ?

Réponse

```
SELECT
    r.reg_nom AS "Régions",
    COUNT(c.contrat_id) AS "Nombre de contrats"
FROM contrat AS c
JOIN region AS r
ON c.code_dep_code_commune = r.code_dep_code_commune
GROUP BY r.reg_nom
ORDER BY COUNT(c.contrat_id) DESC;
```

Résultat

The screenshot shows a PostgreSQL query editor interface. The top bar displays the connection information: 'gestion_logement/postgres@PostgreSQL 18'. Below the toolbar, there are tabs for 'Query' (which is selected), 'Query History', 'Data Output' (selected), 'Messages', and 'Notifications'. The main area contains a SQL query and its results.

```
1 SELECT
2     r.reg_nom AS "Régions",
3     COUNT(c.contrat_id) AS "Nombre de contrats"
4 FROM contrat AS c
5 JOIN region AS r
6 ON c.code_dep_code_commune = r.code_dep_code_commune
7 GROUP BY r.reg_nom
8 ORDER BY COUNT(c.contrat_id) DESC;
```

The results table has two columns: 'Régions' (character varying (50)) and 'Nombre de contrats' (bigint). The data is as follows:

	Régions	Nombre de contrats
1	Ile-de-France	14177
2	Provence-Alpes-Côte d'A...	3279
3	Auvergne-Rhône-Alpes	3042
4	Nouvelle-Aquitaine	2038
5	Occitanie	1609
6	Pays de la Loire	1196
7	Hauts-de-France	1189
8	Bretagne	947
9	Normandie	824

Total rows: 16 Query complete 00:00:00.138 CRLF Ln 8, Col 35