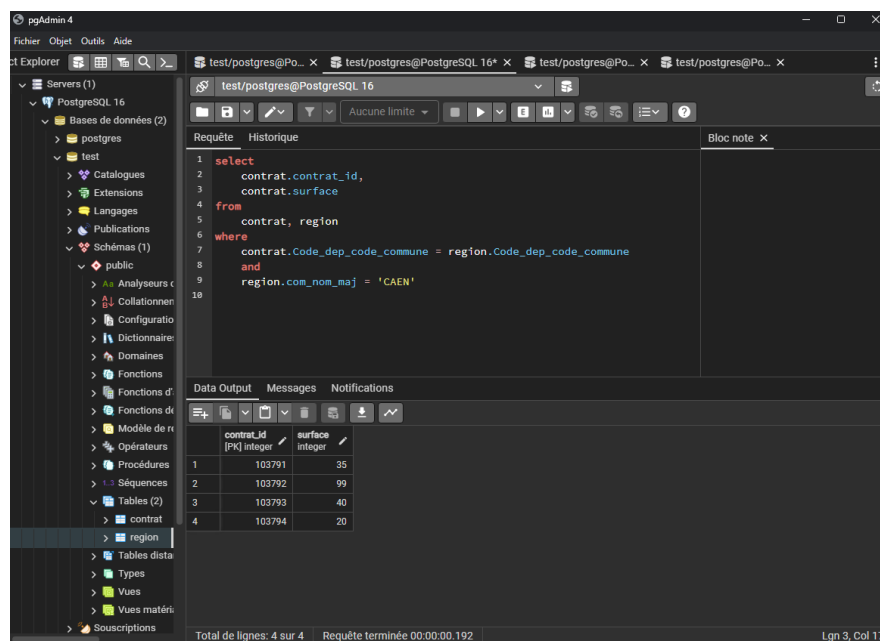


Liste requêtes SQL

Requête 1 : Lister les numéros de contrats (contrat_ID) avec leur surface pour la commune de Caen.

Pour avoir cette information, il faut faire une jointure entre les deux tables car dans la table contrat nous n'avons pas le nom des villes mais uniquement les codes communes. Ensuite nous sélectionnons les colonnes demandées à savoir les numéros de contrat et les surfaces. Pour finir on effectue un filtrage sur le nom de commune pour garder uniquement "CAEN", sans oublier de mettre dans le filtrage la condition de jointure grâce à la clé étrangère présente dans la table contrat qui correspond à la clé primaire de la table région. Cf code SQL et résultat dans l'image ci-dessous.



Requête 2 : Lister les numéros de contrats avec le type de contrat et leur formule pour les maisons du département 71.

Cette fois encore nous allons faire une jointure afin de pouvoir filtrer le département, information qui se trouve uniquement dans la table région. Nous sélectionnons nos colonnes demandées pour répondre à la requête à savoir les numéros de contrat, les types de contrats et les formules. Ensuite nous effectuons un filtrage sur le type de local qui doit être égal à "maison" ainsi que sur le département qui doit être égal au numéro 71. Toujours sans oublier la condition de jointure. CF code SQL et résultat dans l'image ci-dessous.

Requête

```

1 select
2   contrat.contrat_id,
3   contrat.type_contrat,
4   contrat.formule
5 from
6   contrat, region
7 where
8   contrat.code_dep_code_commune = region.code_dep_code_commune
9 and
10  contrat.type_local = 'Maison'
11 and
12  region.dep_code = '71'

```

Data Output

contrat_id [PK] integer	type_contrat character varying	formule character varying
1	114768	Residence principale
2	114779	Residence principale
3	114782	Residence principale
4	114812	Residence principale

Total de lignes: 4 sur 4 Requête terminée 00:00:00.086 Lgn 10, Col 25

Requête 3 : Lister le nom des régions de France.

Ici pas besoin de jointure, car l'information demandée se trouve dans la table région. Nous allons sélectionner la colonne des noms de régions en demandant de ne pas avoir de doublons dans le résultat et de garder ainsi que des informations distinctes. Cela aura pour effet de nous lister les différentes régions de France comme souhaité. Cf code SQL et résultat dans l'image ci-dessous.

pgAdmin 4

FichierObjetOutilsAide

Explorertest/postgres@Po...test/postgres@Po...test/postgres@Po...test/postgres@PostgreSQL 16*test/postgres@PostgreSQL 16

Servers (1)

- PostgreSQL 16
 - Bases de données (2)
 - postgres
 - test
 - Catalogues
 - Extensions
 - Langages
 - Publications
 - Schémas (1)
 - public
 - Analysateurs (c)
 - Collationnemen
 - Configuratio
 - Dictionnaire
 - Domaines
 - Fonctions
 - Fonctions d'
 - Fonctions de
 - Modèle de ri
 - Opérateurs
 - Procédures
 - Séquences
 - Tables (2)
 - contrat
 - region
 - Tables dista
 - Types
 - Vues
 - Vues matérial
 - Souscriptions

test/postgres@PostgreSQL 16

Aucune limite

RequêteHistorique

```
1 select distinct
2   region.reg_nom
3 from
4   region
```

Bloc note x

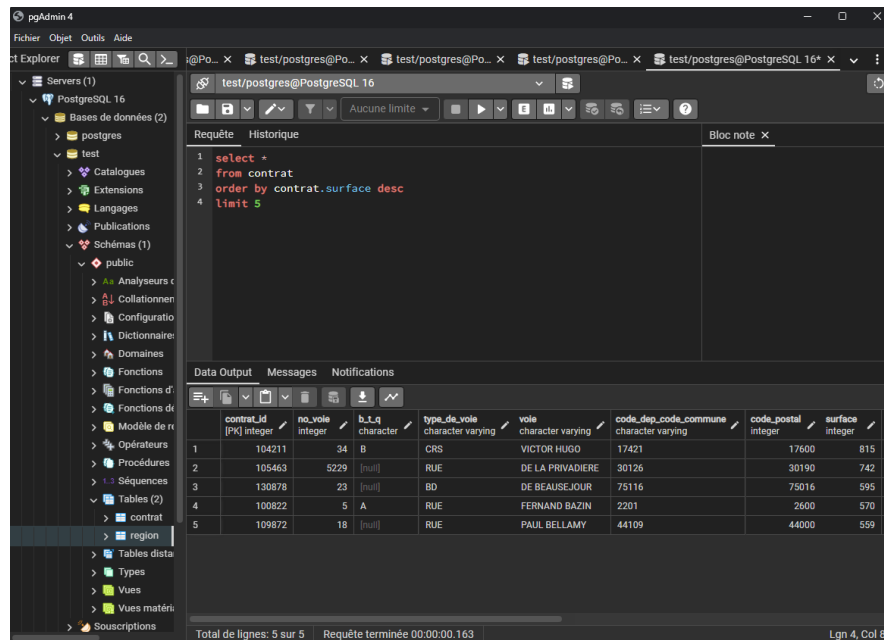
Data OutputMessagesNotifications

reg_nom	character varying
1	Hauts-de-France
2	Île-de-France
3	La Réunion
4	Bretagne
5	Mayotte
6	Auvergne-Rhône-Alpes
7	Grand Est
8	Corse
9	Collectivités d'outre-mer
10	Guyane

Total de lignes: 19 sur 19Requête terminée 00:00:00.099

Requête 4 : Quels sont les 5 contrats qui ont les surfaces les plus élevées ?

Pour cette demande j'ai conservé toutes les colonnes de la table 'contrat'. J'ai ensuite trié par ordre décroissant les valeurs de la colonne 'surface' puis je n'ai gardé que les 5 premières pour avoir le résultat demandé.



The screenshot shows the pgAdmin 4 interface. The left sidebar displays the database structure for 'test/postgres@PostgreSQL 16'. The main window shows a SQL query in the 'Requête' tab:

```
1 select *
2 from contrat
3 order by contrat.surface desc
4 limit 5
```

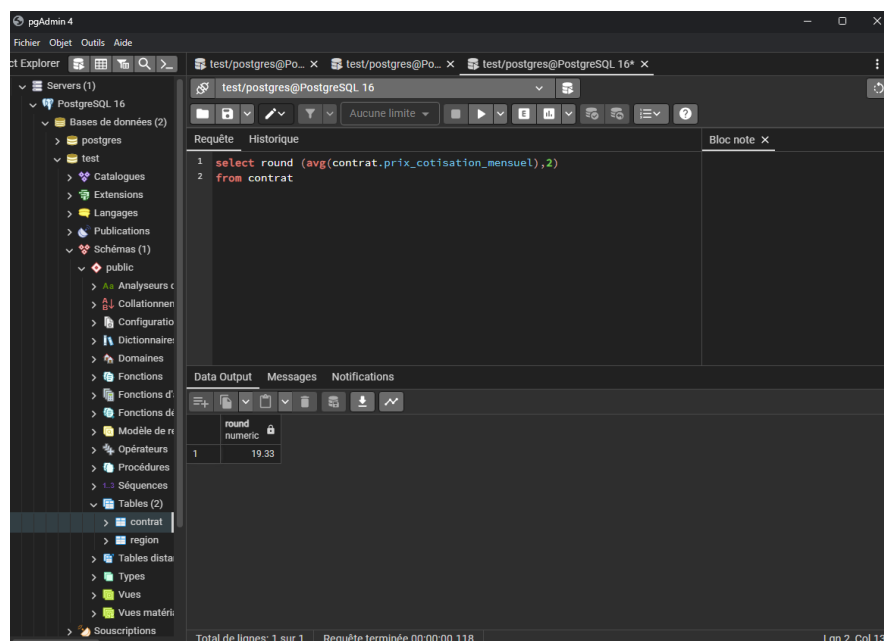
The 'Data Output' tab shows the results of the query as a table with 8 columns: contrat_id, no_vois, b.t.q, type_de_vois, voies, codes_dep_code_commune, codes_postal, and surface. The results are ordered by surface in descending order, showing the top 5 contracts.

contrat_id	no_vois	b.t.q	type_de_vois	voies	codes_dep_code_commune	codes_postal	surface
1	104211	34	B	CRS	VICTOR HUGO	17421	815
2	105463	5229	[null]	RUE	DE LA PRIVADIERE	30126	742
3	130878	23	[null]	BD	DE BEAUX-BOIS	75116	595
4	100822	5	A	RUE	FERNAND BAZIN	2201	570
5	109872	18	[null]	RUE	PAUL BELLAMY	44109	559

Total de lignes: 5 sur 5 Requête terminée 00:00:00, 163 Lgn 4, Col 8

Requête 5 : Quel est le prix moyen de la cotisation mensuelle ?

J'ai besoin ici uniquement de la colonne 'prix_cotisation_mensuel' de la table 'contrat'. J'applique juste la fonction "AVG" (average) pour calculer la moyenne de toutes les valeurs de la colonne que j'ai arrondie à deux chiffres après la virgule pour une meilleure lisibilité et une meilleure exploitation dans le cadre d'analyse business.



The screenshot shows the pgAdmin 4 interface. The left sidebar displays the database structure for 'test/postgres@PostgreSQL 16'. The main window shows a SQL query in the 'Requête' tab:

```
1 select round (avg(contrat.prix_cotisation_mensuel),2)
2 from contrat
```

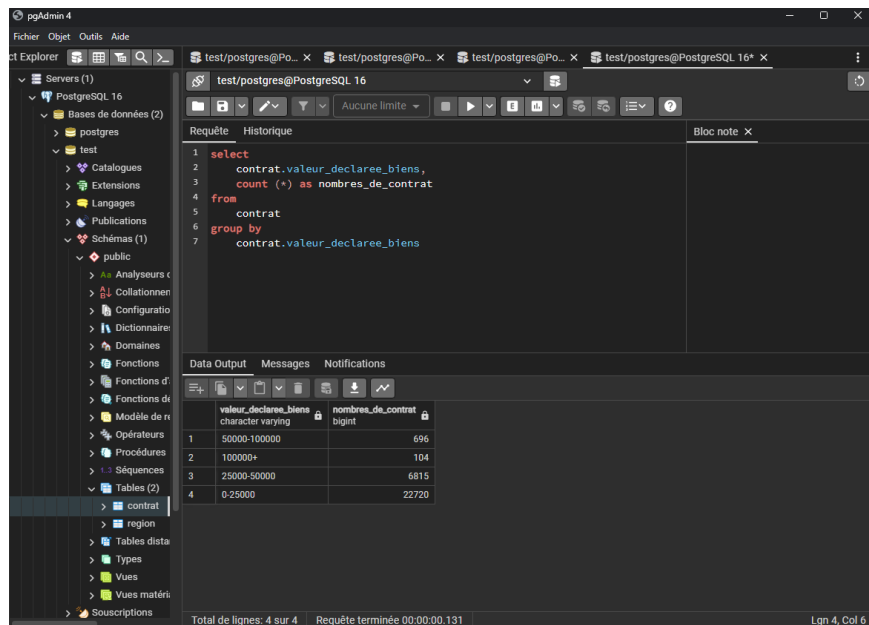
The 'Data Output' tab shows the results of the query as a table with 1 column: round. The result is a single row with the value 19.33.

round
19.33

Total de lignes: 1 sur 1 Requête terminée 00:00:00, 118 Lgn 2, Col 13

Requête 6 : Quel est le nombre de contrats pour chaque catégorie de prix de la valeur déclarée des biens ?

J'ai besoin de faire des agrégats pour chaque valeur de 'valeur_declaree_biens' via un group by et de créer une colonne appelé ici 'nombre_de_contrat' qui représentera la somme du nombre de contrat par groupe d'agrégat créé.



The screenshot shows the pgAdmin 4 interface. On the left, the 'Servers' tree is expanded to show the 'test' database. The 'Tables' folder is selected, and the 'contrat' table is highlighted. The main pane displays a SQL query:

```
1 select
2   contrat.valeur_declaree_biens,
3   count (*) as nombres_de_contrat
4 from
5   contrat
6 group by
7   contrat.valeur_declaree_biens
```

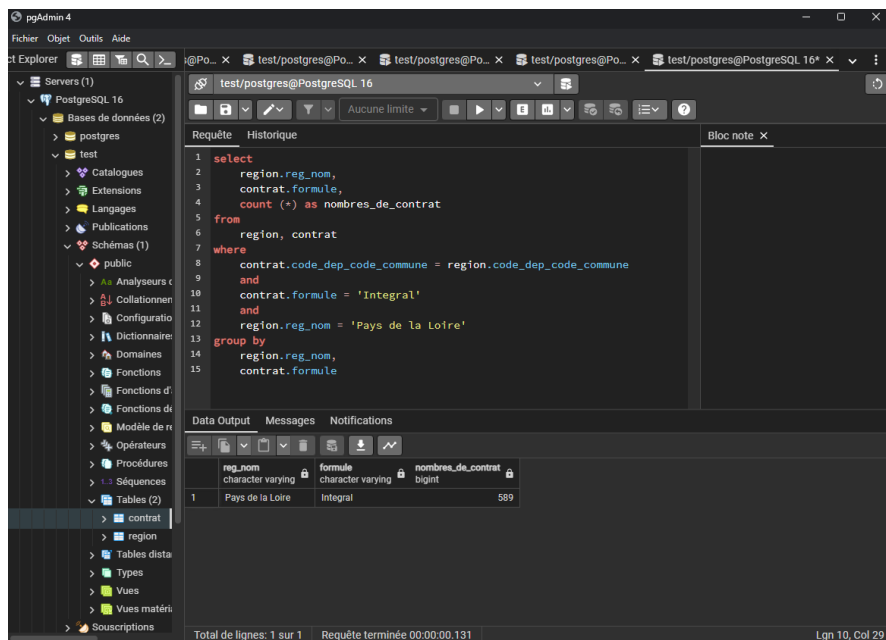
Below the query, the 'Data Output' tab shows the results of the query:

valeur_declaree_biens	nombres_de_contrat
50000-100000	696
100000+	104
25000-50000	6815
0-25000	22720

The status bar at the bottom indicates 'Total de lignes: 4 sur 4' and 'Requête terminée 00:00:00.131'.

Requête 7 : Quel est le nombre de formule “integral” sur la région Pays de la Loire ?

Pour répondre à cette demande, j'ai besoin de faire deux agrégations : une sur la colonne 'formule' de la table 'contrat' et une sur la colonne 'reg_nom' de la table 'region'. Comme sur la requête précédente, j'ajoute une colonne qui donnera le nombre d'occurrences pour chaque groupe créé. Ensuite j'effectue un filtrage avec les valeurs demandées dans la question afin d'obtenir notre résultat.



The screenshot shows the pgAdmin 4 interface. On the left, the 'Servers' tree is expanded to show the 'test' database. The 'Tables' folder is selected, and the 'contrat' table is highlighted. The main pane displays a SQL query:

```
1 select
2   region.reg_nom,
3   contrat.formule,
4   count (*) as nombres_de_contrat
5 from
6   region, contrat
7 where
8   contrat.code_dep_code_commune = region.code_dep_code_commune
9 and
10  contrat.formule = 'Integral'
11 and
12  region.reg_nom = 'Pays de la Loire'
13 group by
14   region.reg_nom,
15   contrat.formule
```

Below the query, the 'Data Output' tab shows the results of the query:

reg_nom	formule	nombres_de_contrat
Pays de la Loire	Integral	589

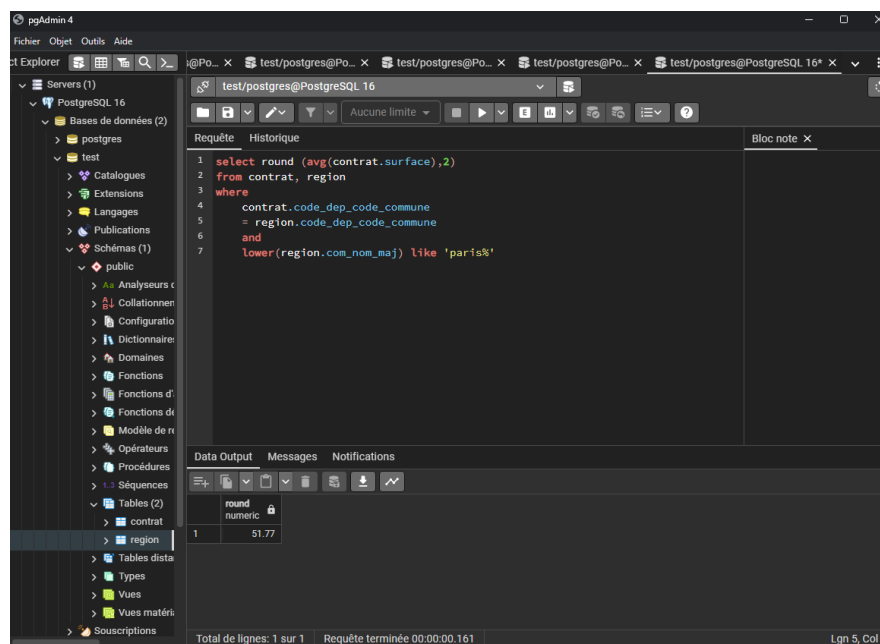
The status bar at the bottom indicates 'Total de lignes: 1 sur 1' and 'Requête terminée 00:00:00.131'.

Requête 8 : Lister les numéros de contrats avec le type de contrat et leur formule pour les maisons du département 71.

Exactement la même requête que la n°2.

Requête 9 : Quelle est la surface moyenne des contrats à Paris ?

Pour avoir cette information, j'ai besoin de la colonne 'surface' de la table 'contrat' sur laquelle je vais effectuer un calcul de moyenne arrondie à 2 chiffres après la virgule. Ensuite j'ai besoin de faire une jointure entre nos deux tables pour effectuer notre filtrage. Ce dernier est effectué en recherchant les occurrences pour la ville 'Paris' grâce à l'attribut **like** 'paris%' afin de prendre en compte tous les arrondissements parisiens. J'ai rajouté la fonction **lower** à la colonne 'com_nom_maj' afin de faciliter la recherche et d'éviter les erreurs. Ce qui nous donne comme résultats : une surface moyenne de 51.77 M² pour les contrats à Paris.



Requête 10 : Classements des 10 départements où le prix moyen de la cotisation est le plus élevé.

J'aurais besoin ici des colonnes des noms de département que je vais agréger et du prix des cotisations sur laquelle je vais calculer la moyenne. J'effectue également une jointure car les informations se trouvent dans nos deux tables. Pour finir je trie la colonne des prix moyens de cotisation par ordre décroissant et je ne garde que les 10 premières valeurs pour obtenir le résultat attendu.

Requête Historique

```

1 select
2   region.dep_nom,
3   round (avg(contrat.prix_cotisation_mensuel),2) as cotisation_mensuel_moyen
4 from
5   contrat, region
6 where
7   contrat.code_dep_code_commune = region.code_dep_code_commune
8 group by
9   region.dep_nom
10 order by
11   cotisation_mensuel_moyen desc
12 limit
13   10

```

Data Output Messages Notifications

	dep_nom character varying	cotisation_mensuel_moyen numeric
1	Paris	36.40
2	Hauts-de-Seine	26.49
3	Val-de-Marne	20.31
4	Yvelines	18.83
5	Rhône	18.49
6	Ain	18.24
7	Alpes-Maritimes	18.14
8	Charente-Maritime	17.32
9	Haute-Savoie	17.15
10	Corse-du-Sud	17.07

Total de lignes: 10 sur 10 Requête terminée 00:00:00.154 Lgn 9, Col 12

Requête 11 : Liste des communes ayant eu au moins 150 contrats.

Je vais avoir une nouvelle fois besoin de faire une jointure afin d'obtenir les informations issues des deux tables. Je vais effectuer une première fonction d'agrégation sur la colonne des noms de commune, puis une deuxième pour calculer la somme des contrats par groupe d'agrégats créés. Je fini cette requête par un filtrage dans les agrégations grâce à **'having'** pour ne garder que les résultats supérieurs ou égaux à 150, et je trie par ordre décroissant la colonne nombre de contrats pour une meilleure lisibilité.

Requête Historique

```

1 select
2   region.com_nom_maj,
3   count(*) as nombres_de_contrat
4 from
5   region, contrat
6 where
7   contrat.code_dep_code_commune = region.code_dep_code_commune
8 group by
9   region.com_nom_maj
10 having
11   count(*) >= 150
12 order by
13   nombres_de_contrat desc

```

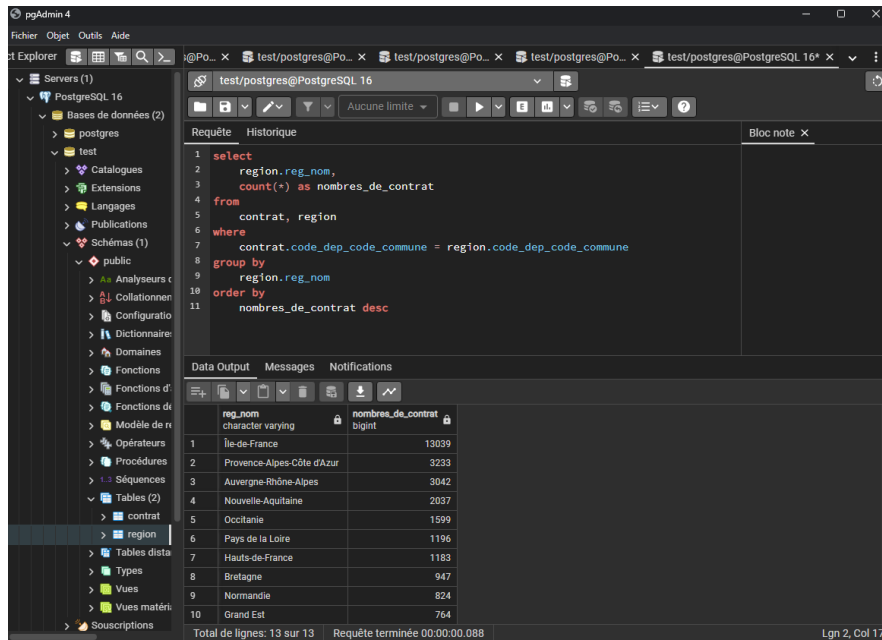
Data Output Messages Notifications

	com_nom_maj character varying	nombres_de_contrat bigint
1	PARIS 18E ARRONDISSEMENT	515
2	PARIS 17E ARRONDISSEMENT	468
3	PARIS 19E ARRONDISSEMENT	407
4	PARIS 16E ARRONDISSEMENT	394
5	NICE	387
6	PARIS 11E ARRONDISSEMENT	381
7	BORDEAUX	302
8	PARIS 20E ARRONDISSEMENT	302
9	NANTES	291
10	PARIS 19E ARRONDISSEMENT	266

Total de lignes: 20 sur 20 Requête terminée 00:00:00.066 Lgn 11, Col 17

Requête 12 : Quel est le nombre de contrats pour chaque région ?

Je suis le même cheminement que la requête précédente : en effectuant une jointure, une agrégation ; cette fois ci sur le nom des régions ; et en rajoutant une colonne calculant le nombre de contrat par région avec la fonction '**count**'. Je n'ai pas besoin d'effectuer de filtrage. Et toujours dans un souci d'une meilleure lisibilité, je trie par ordre décroissant la colonne nombre de contrats.



The screenshot shows the pgAdmin 4 interface. On the left, the 'Servers' tree is expanded to show the 'public' schema. The main pane displays a SQL query in the 'Requête' tab. The query is as follows:

```
1 select
2   region.reg_nom,
3   count(*) as nombres_de_contrat
4 from
5   contrat, region
6 where
7   contrat.code_dep_code_commune = region.code_dep_code_commune
8 group by
9   region.reg_nom
10 order by
11   nombres_de_contrat desc
```

Below the query, the 'Data Output' tab shows the results of the query. The results are displayed in a table with two columns: 'reg_nom' (character varying) and 'nombres_de_contrat' (bigint). The data is sorted in descending order of 'nombres_de_contrat'.

reg_nom	nombres_de_contrat
Île-de-France	13039
Provence-Alpes Côte d'Azur	3233
Auvergne-Rhône-Alpes	3042
Nouvelle-Aquitaine	2037
Occitanie	1599
Pays de la Loire	1196
Hauts-de-France	1183
Bretagne	947
Normandie	824
Grand Est	764

The status bar at the bottom indicates 'Total de lignes: 13 sur 13' and 'Requête terminée 00:00:00.088'.