

## Requêtes SQL

**Requête 1:** Lister les numéros de contrats (contrat\_ID) avec leur surface pour la commune de Caen.

```
SELECT    contrat_ID as numero_contrat,
          Com_nom as Commune,
          Surface
FROM      contrat, region
WHERE     Region_Code_dep_code_commune =
          Code_dep_code_commune
AND       Com_nom = 'CAEN'
ORDER BY Surface
```

The screenshot shows a SQL query editor interface. The query is as follows:

```
1 • SELECT contrat_ID as numero_contrat,
2       Com_nom as Commune,
3       Surface
4 FROM   contrat, region
5 WHERE  Region_Code_dep_code_commune = Code_dep_code_commune
6 AND    Com_nom = "CAEN"
7 ORDER BY Surface
```

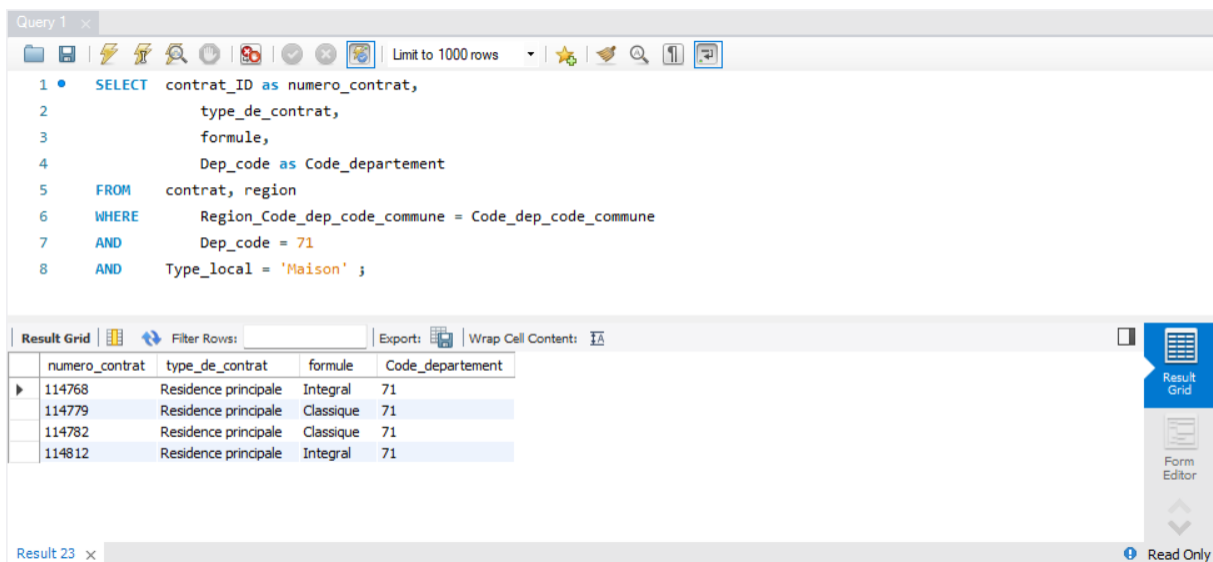
Below the query, the results are displayed in a table with the following data:

	numero_contrat	Commune	Surface
▶	103794	CAEN	20
	103791	CAEN	35
	103793	CAEN	40
	103792	CAEN	99

The interface also includes a toolbar with various icons, a 'Limit to 1000 rows' dropdown, and a 'Result Grid' button on the right side. The bottom status bar indicates 'Result 24' and 'Read Only'.

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

```
SELECT    contrat_ID as numero_contrat,
          type_de_contrat,
          formule,
          Dep_code as Code_departement
FROM      contrat, region
WHERE     Region_Code_dep_code_commune =
          Code_dep_code_commune
AND       Dep_code = 71
AND       Type_local = 'Maison' ;
```



The screenshot shows a database query tool interface. The top section displays the SQL query, and the bottom section shows the results in a grid. The query is as follows:

```
1 • SELECT contrat_ID as numero_contrat,
2         type_de_contrat,
3         formule,
4         Dep_code as Code_departement
5 FROM   contrat, region
6 WHERE  Region_Code_dep_code_commune = Code_dep_code_commune
7 AND    Dep_code = 71
8 AND    Type_local = 'Maison' ;
```

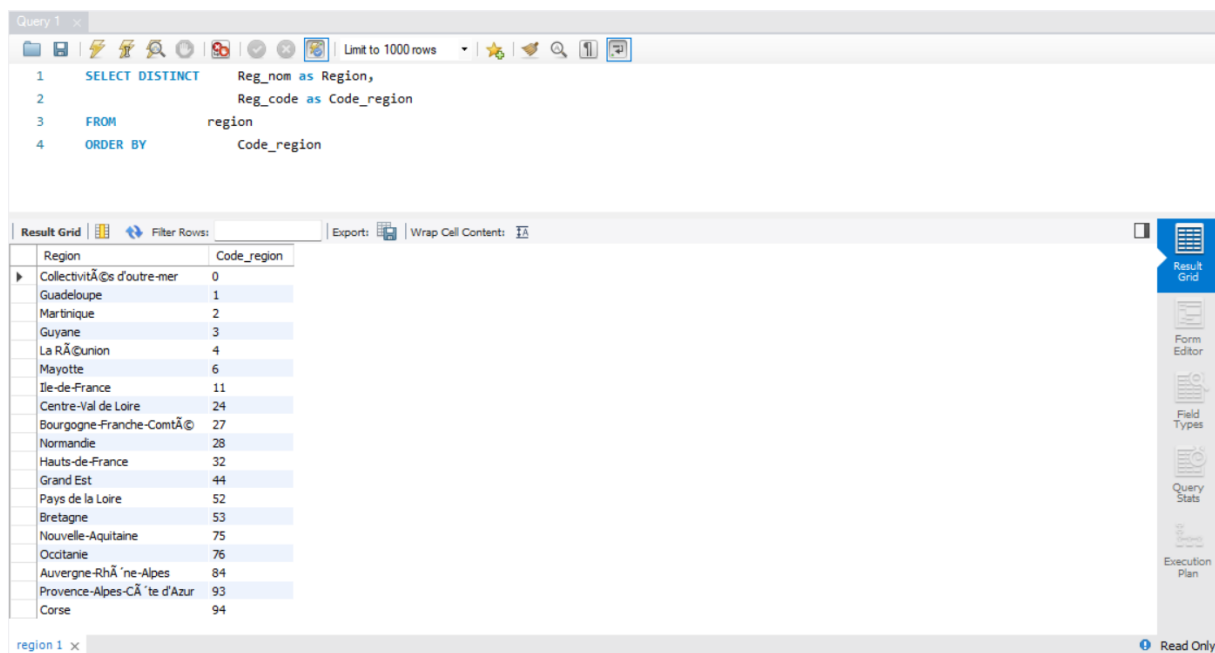
The results grid shows the following data:

	numero_contrat	type_de_contrat	formule	Code_departement
▶	114768	Residence principale	Integral	71
	114779	Residence principale	Classique	71
	114782	Residence principale	Classique	71
	114812	Residence principale	Integral	71

The interface also includes a toolbar with various icons, a 'Limit to 1000 rows' dropdown, and a 'Read Only' status indicator at the bottom right.

**Requête 3 :** Lister le nom des régions de France.

```
SELECT DISTINCT      Reg_nom as Region
                     Code_region
FROM                  region
ORDER BY              Code_region
```



The screenshot shows a database query editor with a query named 'Query 1'. The query is as follows:

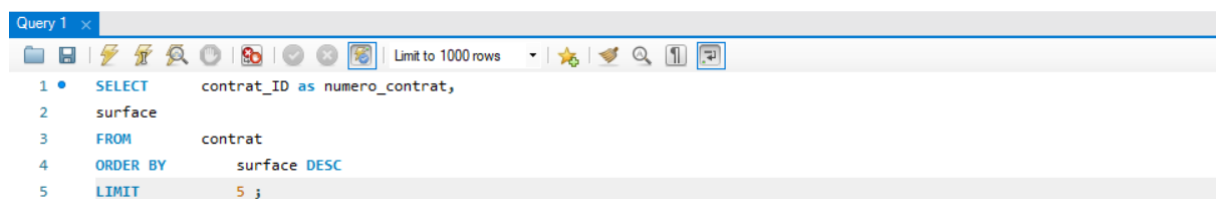
```
1 SELECT DISTINCT      Reg_nom as Region,
2                     Reg_code as Code_region
3 FROM                  region
4 ORDER BY              Code_region
```

Below the query editor, the 'Result Grid' is displayed, showing the results of the query. The grid has two columns: 'Region' and 'Code\_region'. The results are as follows:

Region	Code_region
Collectivités d'outre-mer	0
Guadeloupe	1
Martinique	2
Guyane	3
La Réunion	4
Mayotte	6
Ile-de-France	11
Centre-Val de Loire	24
Bourgogne-Franche-Comté	27
Normandie	28
Hauts-de-France	32
Grand Est	44
Pays de la Loire	52
Bretagne	53
Nouvelle-Aquitaine	75
Occitanie	76
Auvergne-Rhône-Alpes	84
Provence-Alpes-Côte d'Azur	93
Corse	94

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

```
SELECT      contrat_ID as numero_contrat,  
            Surface  
FROM        contrat  
ORDER BY    surface DESC  
LIMIT       5 ;
```



The screenshot shows the results of the query in a table format. The table has two columns: "numero\_contrat" and "surface". The results are as follows:

numero_contrat	surface
104211	815
105463	742
130878	595
100822	570
109872	559

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

```
SELECT    round(avg(prix_cotisation), 2) as Moyenne_cotisation  
FROM      contrat
```

The screenshot shows a database query tool interface. At the top, a tab labeled 'Query 1' is active. Below it is a toolbar with various icons for file operations, execution, and viewing. The SQL query is entered in a text area:

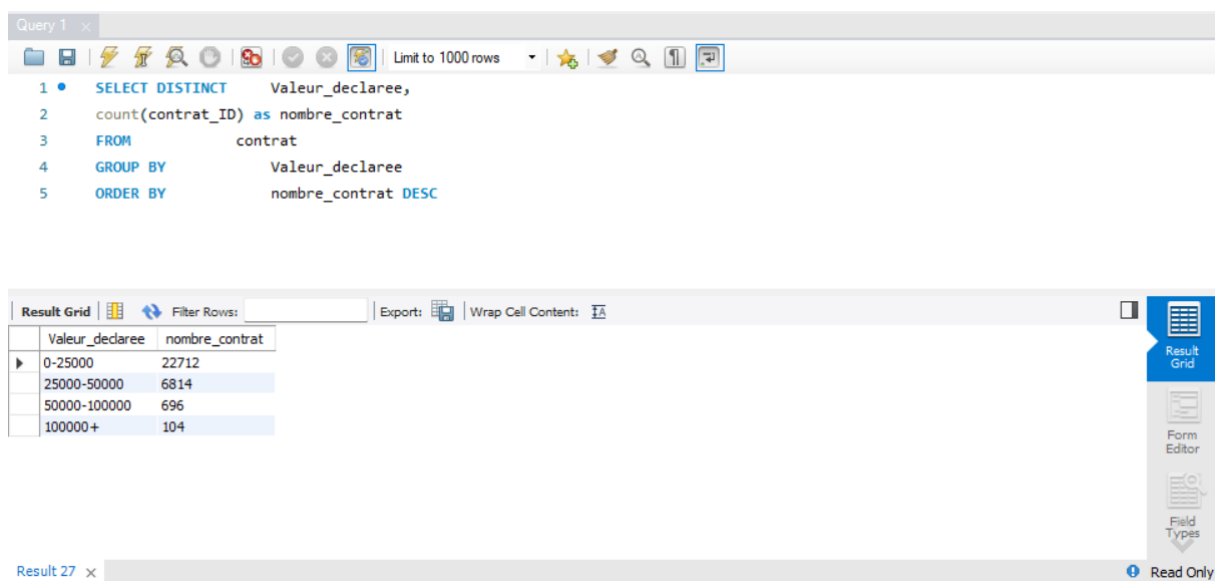
```
1 • SELECT round(avg(prix_cotisation), 2) as Moyenne_cotisation  
2 FROM contrat
```

Below the query editor, there is a 'Result Grid' section. It includes a 'Filter Rows' field, an 'Export' button, and a 'Wrap Cell Content' checkbox. The result grid displays a single row with the column header 'Moyenne\_cotisation' and the value '19.33'. On the right side of the interface, there is a vertical toolbar with buttons for 'Result Grid', 'Form Editor', and a 'Read Only' status indicator.

Moyenne_cotisation
19.33

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

```
SELECT DISTINCT      Valeur_declaree,
                     count(*) as nombre_contrat
FROM                  contrat
GROUP BY              Valeur_declaree
ORDER BY              nombre_contrat DESC
```



The screenshot shows a database query tool interface. The top section displays the SQL query for 'Query 1'. The query is: `SELECT DISTINCT Valeur_declaree, count(contract_ID) as nombre_contrat FROM contrat GROUP BY Valeur_declaree ORDER BY nombre_contrat DESC`. The bottom section shows the 'Result Grid' with the following data:

Valeur_declaree	nombre_contrat
0-25000	22712
25000-50000	6814
50000-100000	696
100000+	104

The interface includes a toolbar with various icons, a 'Limit to 1000 rows' dropdown, and a 'Read Only' status indicator at the bottom right.

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

```
SELECT      Reg_nom as Region,

            formule,

            count(Contrat_ID) as nombre_contrat

FROM        contrat

JOIN        region

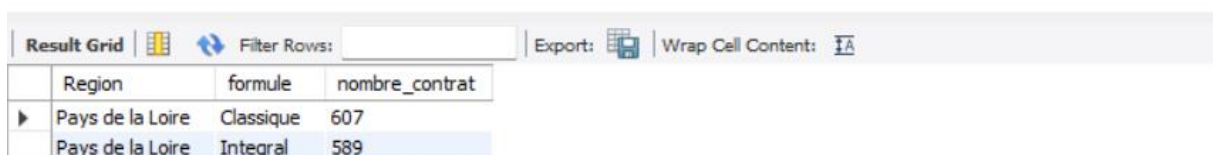
ON          Region_Code_dep_code_commune =
            Code_dep_code_commune

GROUP BY    formule, Reg_nom

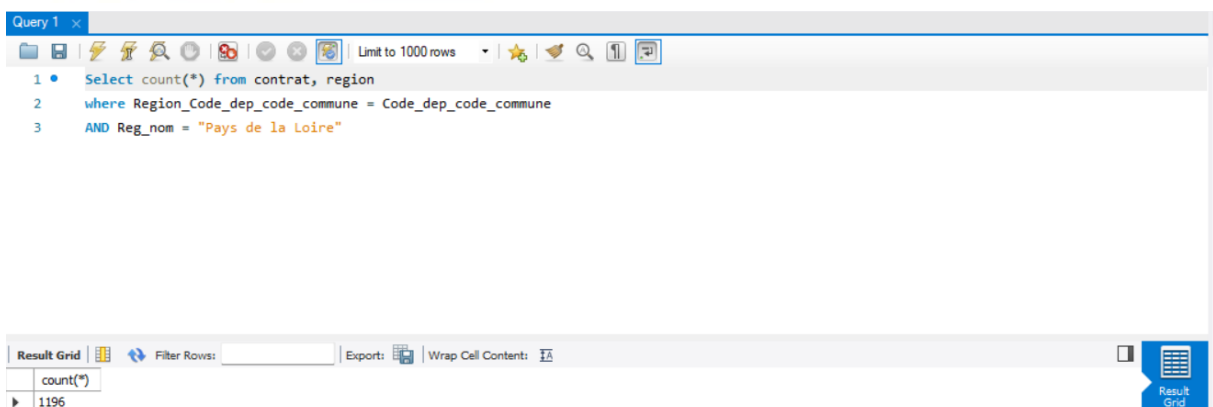
HAVING      Formule = "integral" AND Reg_nom = "Pays de la Loire"
```



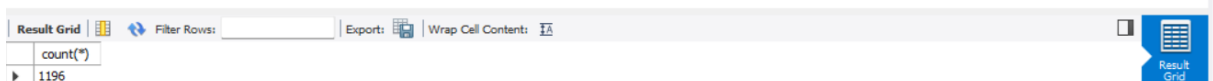
```
1 • SELECT Reg_nom as Region,
2         formule,
3         count(Contrat_ID) as nombre_contrat
4 FROM   contrat
5 JOIN   region
6 ON     Region_Code_dep_code_commune = Code_dep_code_commune
7 GROUP BY formule, Reg_nom
8 HAVING Reg_nom = "Pays de la Loire"
```



Region	formule	nombre_contrat
Pays de la Loire	Classique	607
Pays de la Loire	Integral	589



```
1 • Select count(*) from contrat, region
2 where Region_Code_dep_code_commune = Code_dep_code_commune
3 AND Reg_nom = "Pays de la Loire"
```



count(*)
1196

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

```
SELECT      Contrat_ID as numero_contrat,

            Type_de_contrat,

            Formule,

            Dep_code as Code_departement

FROM        contrat

JOIN        region

ON          Region_Code_dep_code_commune =
            Code_dep_code_commune

GROUP BY    Contrat_ID,

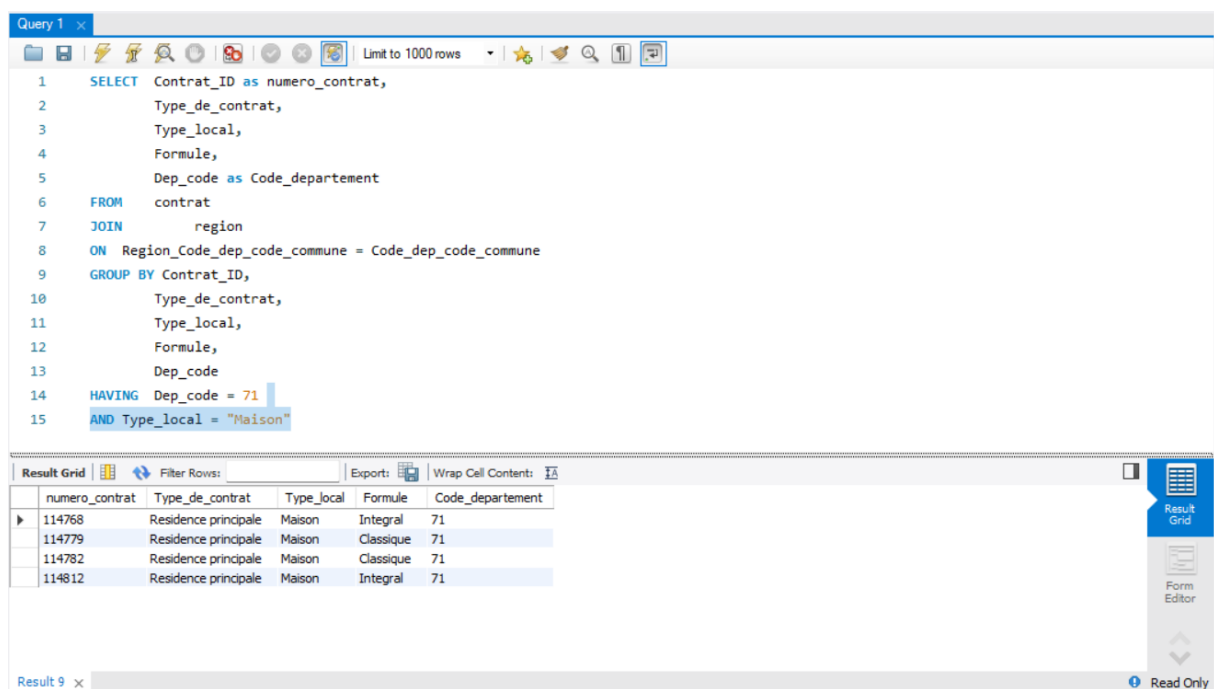
            Type_de_contrat,

            Formule,

            Dep_code

HAVING      Dep_code = 71

AND Type_local = 'Maison'
```



The screenshot shows a database query editor window titled 'Query 1'. The SQL query is displayed in the main area, and the results are shown in a table below. The table has five columns: numero\_contrat, Type\_de\_contrat, Type\_local, Formule, and Code\_departement. The results show four rows of data for contracts in department 71, all of which are 'Maison' type.

numero_contrat	Type_de_contrat	Type_local	Formule	Code_departement
114768	Residence principale	Maison	Integral	71
114779	Residence principale	Maison	Classique	71
114782	Residence principale	Maison	Classique	71
114812	Residence principale	Maison	Integral	71



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

```
SELECT    Dep_nom as Departement,

          Round(avg(Surface), 2) as Moyenne_surface

FROM      contrat

JOIN      region

ON        Region_Code_dep_code_commune =
          Code_dep_code_commune

GROUP BY  Dep_nom

HAVING    Dep_nom = "Paris"
```

The screenshot shows a SQL query editor window titled "Query 1" with a toolbar and a list of icons. The query text is as follows:

```
1 • SELECT Dep_nom as Departement,
2         Round(avg(Surface), 2) as Moyenne_surface
3 FROM    contrat
4 JOIN    region
5 ON      Region_Code_dep_code_commune = Code_dep_code_commune
6 GROUP BY Dep_nom
7 HAVING  Dep_nom = "Paris"
```

Below the query editor is a "Result Grid" section. It includes a "Filter Rows:" input field, an "Export:" button, and a "Wrap Cell Content:" checkbox. The results table has two columns: "Departement" and "Moyenne\_surface". The first row shows "Paris" with a value of "51.77".

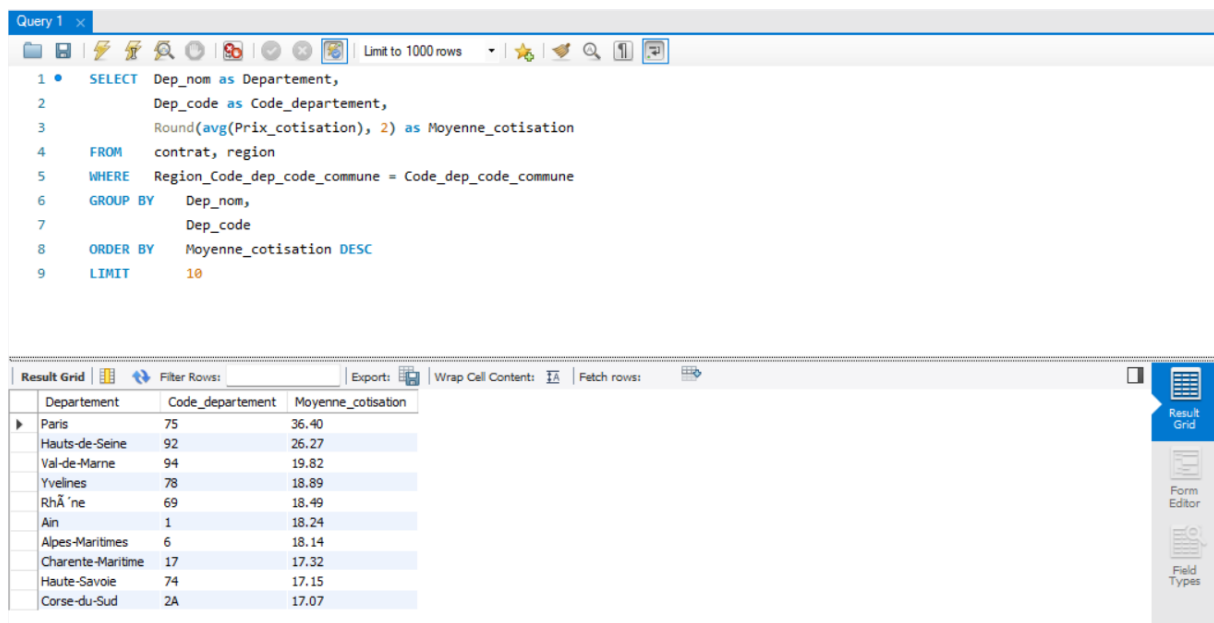
Departement	Moyenne_surface
Paris	51.77

On the right side of the interface, there are buttons for "Result Grid", "Form Editor", and "Read Only".

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

```
SELECT      Dep_nom as Departement,
            Dep_code as Code_departement,
            Round(avg(Prix_cotisation), 2) as Moyenne_cotisation
FROM        contrat, region
WHERE       Region_Code_dep_code_commune =
            Code_dep_code_commune

GROUP BY    Dep_nom
ORDER BY    Moyenne_cotisation DESC
LIMIT      10
```



The screenshot shows a database query editor with a query window at the top and a result grid at the bottom. The query window contains the following SQL code:

```
1 SELECT Dep_nom as Departement,
2       Dep_code as Code_departement,
3       Round(avg(Prix_cotisation), 2) as Moyenne_cotisation
4 FROM   contrat, region
5 WHERE  Region_Code_dep_code_commune = Code_dep_code_commune
6 GROUP BY Dep_nom,
7          Dep_code
8 ORDER BY Moyenne_cotisation DESC
9 LIMIT 10
```

The result grid displays the following data:

Departement	Code_departement	Moyenne_cotisation
Paris	75	36.40
Hauts-de-Seine	92	26.27
Val-de-Marne	94	19.82
Yvelines	78	18.89
Rhône	69	18.49
Ain	1	18.24
Alpes-Maritimes	6	18.14
Charente-Maritime	17	17.32
Haute-Savoie	74	17.15
Corse-du-Sud	2A	17.07

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

```
SELECT      Com_nom as Commune,
            count(contrat_ID) as nombre_contrat

FROM        contrat, region

WHERE       Region_code_dep_code_commune =
            Code_dep_code_commune

GROUP BY    Com_nom

HAVING      count(contrat_ID) > 150

ORDER BY    nombre_contrat;
```

Query 1

Limit to 1000 rows

```
1 SELECT Com_nom as Commune,
2       count(contrat_ID) as nombre_contrat
3 FROM   contrat, region
4 WHERE  Region_code_dep_code_commune = Code_dep_code_commune
5 GROUP BY Com_nom
6 HAVING count(contrat_ID) > 150
7 ORDER BY nombre_contrat;
```

Result Grid

Filter Rows: Export: Wrap Cell Content:

Commune	nombre_contrat
PARIS 3	159
LILLE	161
COURBEVOIE	163
TOULON	170
TOULOUSE	187
PARIS 9	204
GRENOBLE	220
PARIS 14	222
PARIS 12	252
PARIS 10	263
PARIS 19	266
NANTES	291
BORDEAUX	302
PARIS 20	302
PARIS 11	381
NICE	387
PARIS 16	394
PARIS 15	407
PARIS 17	468
PARIS 18	515

Result Grid  
Form Editor  
Field Types  
Query Stats  
Execution Plan

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

```
SELECT    DISTINCT Reg_nom as Region,
          count(Contrat_ID) as nombre_contrat

FROM      contrat, region

WHERE     Region_Code_dep_code_commune =
          Code_dep_code_commune

GROUP BY  Reg_nom

ORDER BY  nombre_contrat ;
```

Query 1

Limit to 1000 rows

```
1 • SELECT DISTINCT Reg_nom as Region,
2       count(Contrat_ID) as nombre_contrat
3 FROM   contrat, region
4 WHERE  Region_Code_dep_code_commune = Code_dep_code_commune
5 GROUP BY Reg_nom
6 ORDER BY nombre_contrat ;
```

Result Grid

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

Region	nombre_contrat
La Réunion	8
Guyane	37
Martinique	73
Corse	247
Bourgogne-Franche-Comté	293
Centre-Val de Loire	598
Grand Est	769
Normandie	824
Bretagne	947
Hauts-de-France	1189
Pays de la Loire	1196
Occitanie	1609
Nouvelle-Aquitaine	2038
Auvergne-Rhône-Alpes	3042
Provence-Alpes-Côte d'Azur	3279
Ile-de-France	14177

Result Grid  
Form Editor  
Field Types  
Query Stats  
Execution Plan