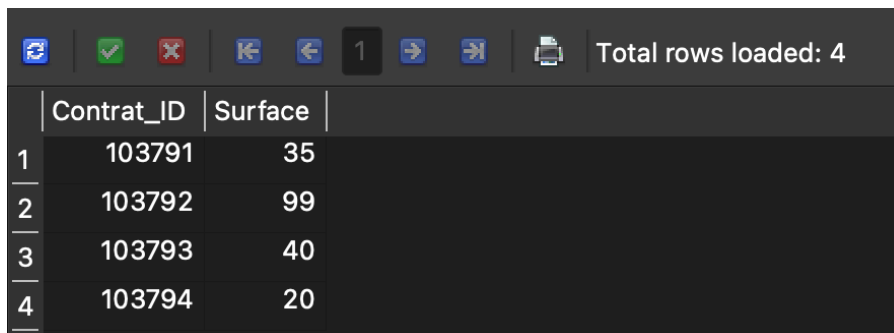


## Requêtes SQL

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

```
SELECT c.Contrat_ID, c.Surface
FROM Contrat AS c
LEFT JOIN Region AS r
ON c.Code_dep_code_commune = r.Code_dep_code_commune
WHERE r.com_nom_maj_court = 'CAEN';
```

Résultat SQLite :

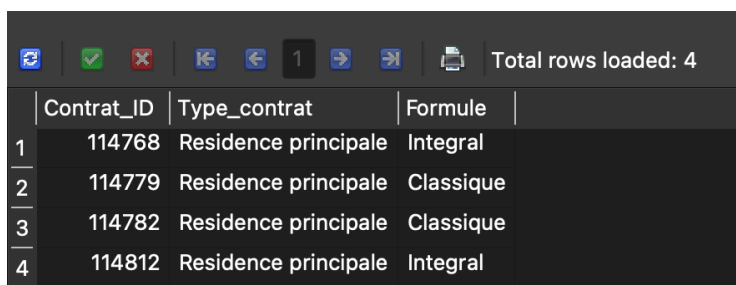
A screenshot of a SQLite query result window. The window has a toolbar at the top with icons for refresh, check, error, first, previous, page 1, next, last, and print. To the right of the toolbar, it says "Total rows loaded: 4". Below the toolbar is a table with two columns: "Contrat\_ID" and "Surface". The table contains four rows of data.

	Contrat_ID	Surface
1	103791	35
2	103792	99
3	103793	40
4	103794	20

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

```
SELECT c.contrat_ID, c.Type_contrat, c.Formule
FROM Contrat AS c
LEFT JOIN Region AS r
ON c.Code_dep_code_commune = r.Code_dep_code_commune
WHERE lower(c.type_local) LIKE 'maison%'
AND r.dep_code = '71';
```

Résultat SQLite :

A screenshot of a SQLite query result window. The window has a toolbar at the top with icons for refresh, check, error, first, previous, page 1, next, last, and print. To the right of the toolbar, it says "Total rows loaded: 4". Below the toolbar is a table with three columns: "Contrat\_ID", "Type\_contrat", and "Formule". The table contains four rows of data.

	Contrat_ID	Type_contrat	Formule
1	114768	Residence principale	Integral
2	114779	Residence principale	Classique
3	114782	Residence principale	Classique
4	114812	Residence principale	Integral

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

```
SELECT DISTINCT reg_nom AS Region  
FROM Region
```

*Résultat SQLite :*

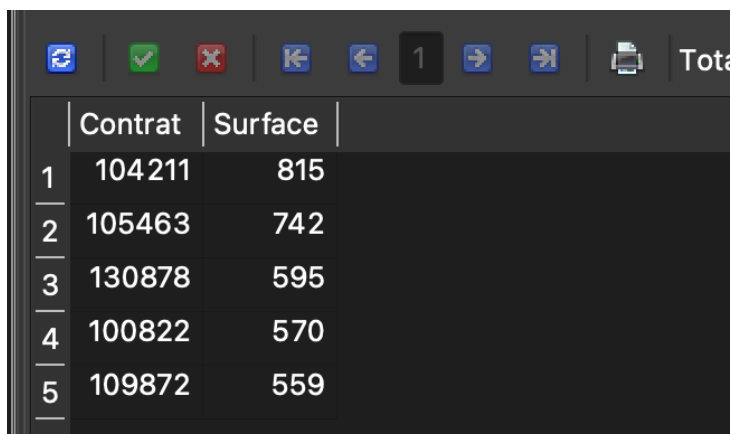


The screenshot shows the SQLite command line interface with a query executed. The results are displayed in a table with two columns: 'Region' and an empty column. The regions listed are: Provence-Alpes-Côte d'Azur, Grand Est, Occitanie, Normandie, Nouvelle-Aquitaine, Centre-Val de Loire, Bourgogne-Franche-Comté, Bretagne, Pays de la Loire, Ile-de-France, Guadeloupe, Martinique, Guyane, La Réunion, Collectivités d'outre-mer, and Mayotte.

	Region	
3	Provence-Alpes-Côte d'Azur	
4	Grand Est	
5	Occitanie	
6	Normandie	
7	Nouvelle-Aquitaine	
8	Centre-Val de Loire	
9	Bourgogne-Franche-Comté	
10	Bretagne	
11	Pays de la Loire	
12	Ile-de-France	
13	Guadeloupe	
14	Martinique	
15	Guyane	
16	La Réunion	
17	Collectivités d'outre-mer	
18	Mayotte	

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

```
SELECT Contrat_ID AS Contrat, Surface  
FROM Contrat  
ORDER BY Surface DESC  
LIMIT 5;
```

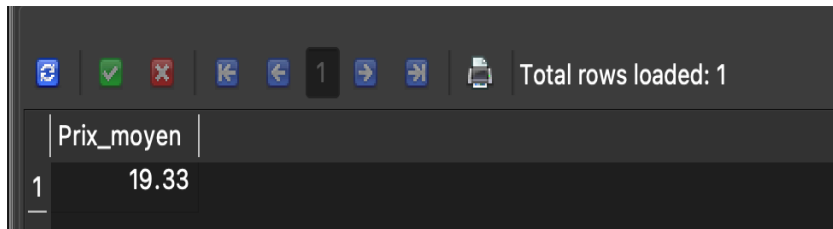


The screenshot shows the SQLite command line interface with a query executed. The results are displayed in a table with three columns: 'Contrat', 'Surface', and an empty column. The top 5 contracts by surface area are listed.

	Contrat	Surface	
1	104211	815	
2	105463	742	
3	130878	595	
4	100822	570	
5	109872	559	

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

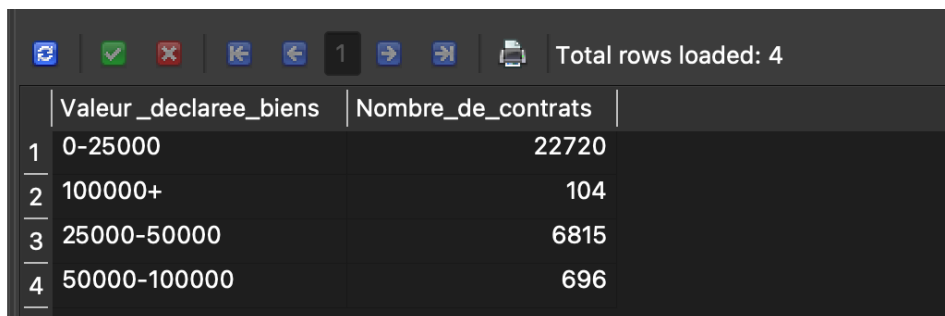
```
SELECT ROUND(AVG(Prix_cotisation_mensuel), 2) AS Prix_moyen  
FROM Contrat;
```



	Prix_moyen
1	19.33

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

```
SELECT valeur_declaree_biens, COUNT(Contrat_ID) AS  
Nombre_de_contrats  
FROM Contrat  
GROUP BY valeur_declaree_biens;
```

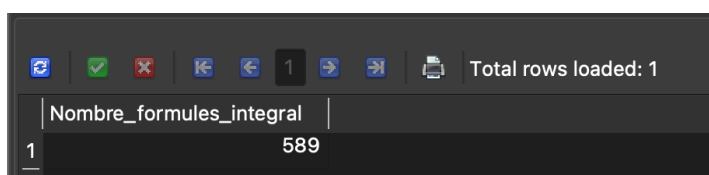


	Valeur_declaree_biens	Nombre_de_contrats
1	0-25000	22720
2	100000+	104
3	25000-50000	6815
4	50000-100000	696

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

```
SELECT COUNT(c.Contrat_ID) AS Nombre_formules_integral  
FROM Contrat c  
LEFT JOIN Region r  
ON c.Code_dep_code_commune = r.Code_dep_code_commune  
WHERE lower(c.Formule) = 'integral'  
AND lower(r.reg_nom) = 'pays de la loire';
```

Résultat SQLite :



	Nombre_formules_integral
1	589

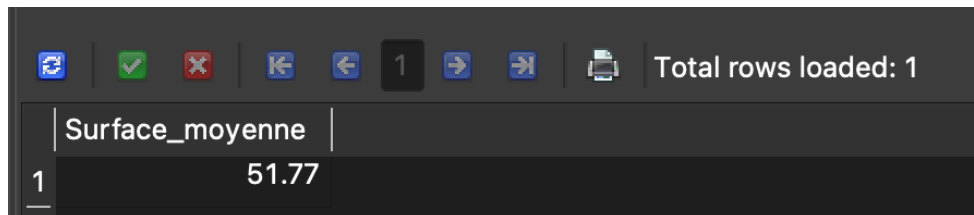
**Requête 8 : Lister les numéros de contrats avec le type de contrat et leur formule pour les maisons du département 71.**

**MÊME DEMANDE QUE REQUÊTE 2.**

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

```
SELECT ROUND(AVG(c.Surface), 2) AS Surface_moyenne
FROM Contrat AS c
LEFT JOIN Region AS r
ON c.Code_dep_code_commune = r.Code_dep_code_commune
WHERE r.com_nom_maj_court LIKE 'PARIS%'
AND r.dep_code = '75';
```

Résultat SQLite :



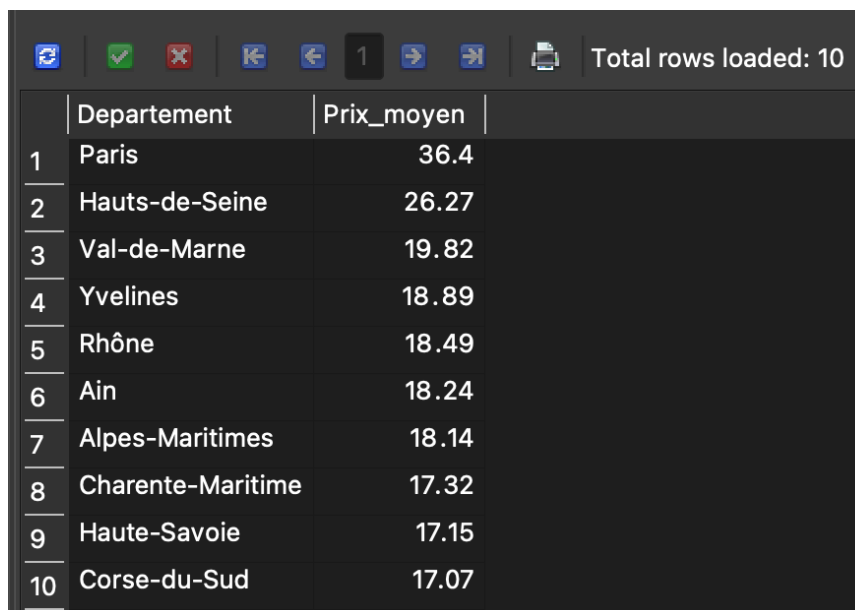
The screenshot shows a SQLite query result interface. At the top, there is a toolbar with icons for refresh, success, error, back, forward, and a page indicator showing '1'. To the right of the toolbar, it says 'Total rows loaded: 1'. Below the toolbar is a table with one column named 'Surface\_moyenne' and one row with the value '51.77'.

	Surface_moyenne
1	51.77

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

```
SELECT r.dep_nom AS Departement,  
ROUND(AVG(c.Prix_cotisation_mensuel), 2) AS Prix_moyen  
FROM Contrat AS c  
LEFT JOIN Region AS r  
ON c.Code_dep_code_commune = r.Code_dep_code_commune  
GROUP BY r.dep_nom  
ORDER BY Prix_moyen DESC  
LIMIT 10;
```

Résultat SQLite :



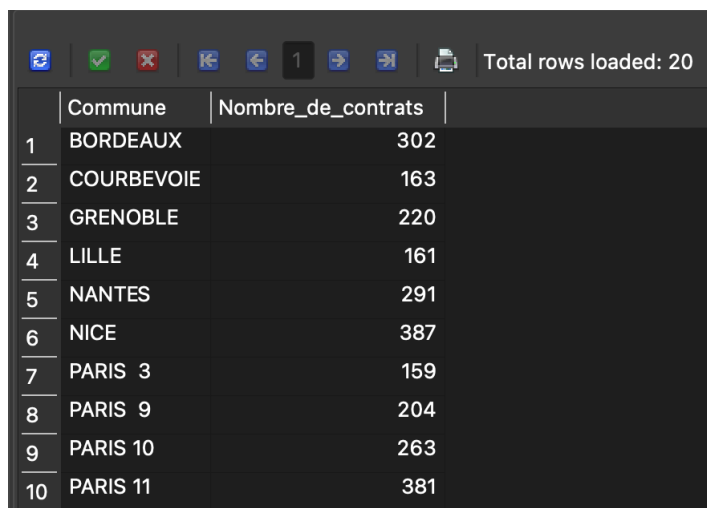
The screenshot shows a SQLite query result interface. At the top, there is a toolbar with various icons (refresh, check, error, back, forward, search, print) and a status bar indicating 'Total rows loaded: 10'. Below the toolbar is a table with two columns: 'Departement' and 'Prix\_moyen'. The table contains 10 rows of data, numbered 1 to 10 in the first column. The departments are listed in descending order of their average monthly subscription price.

	Departement	Prix_moyen
1	Paris	36.4
2	Hauts-de-Seine	26.27
3	Val-de-Marne	19.82
4	Yvelines	18.89
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

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

```
SELECT r.com_nom_maj_court AS Commune, COUNT(c.Contrat_ID) AS  
Nombre_de_contrats  
FROM Contrat AS c  
LEFT JOIN Region AS r  
ON c.Code_dep_code_commune = r.Code_dep_code_commune  
GROUP BY r.com_nom_maj_court  
HAVING COUNT(c.Contrat_ID) >= 150;
```

Résultat SQLite :



	Commune	Nombre_de_contrats
1	BORDEAUX	302
2	COURBEVOIE	163
3	GRENOBLE	220
4	LILLE	161
5	NANTES	291
6	NICE	387
7	PARIS 3	159
8	PARIS 9	204
9	PARIS 10	263
10	PARIS 11	381

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

```
SELECT r.reg_nom AS Region, COUNT(c.Contrat_ID) AS  
Nombre_de_contrats  
FROM Contrat AS c  
LEFT JOIN Region AS r  
  ON c.Code_dep_code_commune = r.Code_dep_code_commune  
GROUP BY r.reg_nom  
ORDER BY Nombre_de_contrats DESC;
```

Résultat SQLite :

	Region	Nombre_de_contrats
1	NULL	9
2	Auvergne-Rhône-Alpes	3042
3	Bourgogne-Franche-Comté	293
4	Bretagne	947
5	Centre-Val de Loire	598
6	Corse	247
7	Grand Est	769
8	Guyane	37
9	Hauts-de-France	1189
10	Ile-de-France	14177
11	La Réunion	8
12	Martinique	73
13	Normandie	824
14	Nouvelle-Aquitaine	2038
15	Occitanie	1609
16	Pays de la Loire	1196
17	Provence-Alpes-Côte d'Azur	3279

Pour connaître les Contrat\_ID sans Region :

```
1 SELECT c.Contrat_ID, c.Code_dep_code_commune, r.reg_nom  
2 FROM Contrat AS c  
3 LEFT JOIN Region AS r  
4   ON c.Code_dep_code_commune = r.Code_dep_code_commune  
5 WHERE r.reg_nom IS NULL
```

Total rows loaded: 9

	Contrat_ID	Code_dep_code_commune	reg_nom
1	128054	97460	NULL
2	128056	97434	NULL
3	128059	97470	NULL
4	128061	97460	NULL
5	128064	97434	NULL
6	128068	97434	NULL
7	128070	97434	NULL
8	128077	97460	NULL
9	128082	97460	NULL