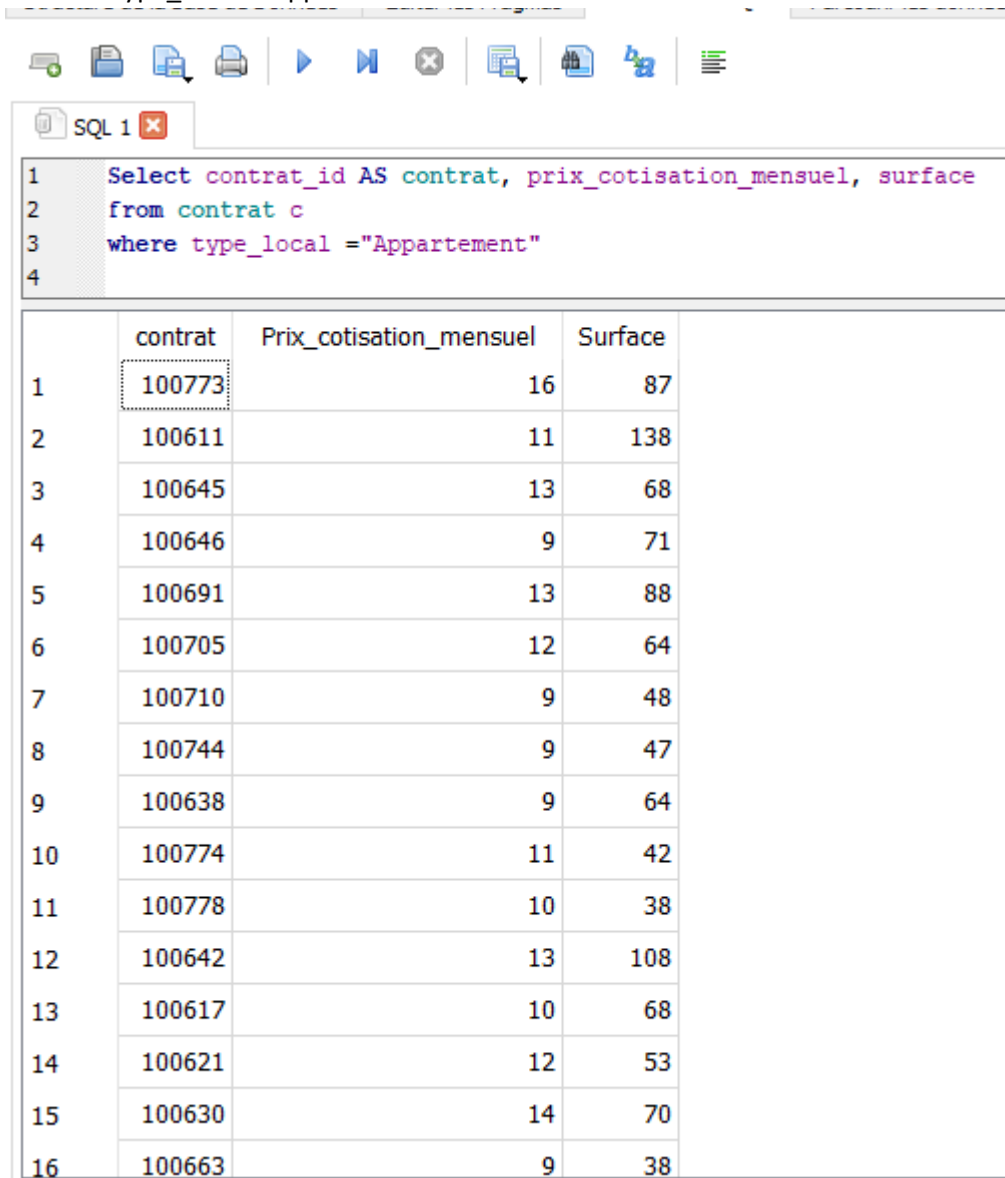




### Requêtes et résultats associés

- 1) Select contrat\_id AS contrat, prix\_cotisation\_mensuel, surface  
from contrat c  
where type\_local = 'Appartement'



The screenshot shows a SQL query editor with a toolbar at the top containing icons for file operations, execution, and formatting. Below the toolbar, a tab labeled 'SQL 1' is active. The query text is as follows:

```
1 Select contrat_id AS contrat, prix_cotisation_mensuel, surface
2 from contrat c
3 where type_local = "Appartement"
4
```

Below the query editor, the results are displayed in a table with 4 columns: an index, 'contrat', 'Prix\_cotisation\_mensuel', and 'Surface'. The table contains 16 rows of data.

	contrat	Prix_cotisation_mensuel	Surface
1	100773	16	87
2	100611	11	138
3	100645	13	68
4	100646	9	71
5	100691	13	88
6	100705	12	64
7	100710	9	48
8	100744	9	47
9	100638	9	64
10	100774	11	42
11	100778	10	38
12	100642	13	108
13	100617	10	68
14	100621	12	53
15	100630	14	70
16	100663	9	38

- 2) SELECT Contrat\_ID as contrat, Surface, Commune  
FROM Contrat  
WHERE Commune="CAEN" ;



Structure de la Base de Données | Éditer les Pragma | Exécuter le SQL

SQL 1

```
1 SELECT Contrat_ID as contrat, Surface, Commune
2 FROM Contrat
3 WHERE Commune="CAEN"
4
```

	contrat	Surface	Commune
1	103791	35	CAEN
2	103792	99	CAEN
3	103793	40	CAEN
4	103794	20	CAEN

L'exécution s'est terminée sans erreur.  
Résultat : 4 enregistrements ramenés en 30ms  
À la ligne 1 :  
SELECT Contrat\_ID as contrat, Surface, Commune  
FROM Contrat  
WHERE Commune="CAEN"

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

```
Select contrat_ID as contrat, type_contrat, formule
from Contrat+
  where Type_local="Maison"
AND Code_département=71 ;
```



SQL 1			
1	Select contrat_ID as contrat, type_contrat, formule		
2	from Contrat		
3	where Type_local="Maison"		
4	AND Code_departement=71		
	contrat	Type_contrat	Formule
1	114768	Residence principale	Integral
2	114782	Residence principale	Classique
3	114812	Residence principale	Integral
4	114779	Residence principale	Classique
L'exécution s'est terminée sans erreur.			
Résultat : 4 enregistrements ramenés en 43ms			
À la ligne 1 :			
Select contrat_ID as contrat, type_contrat, formule			
from Contrat			
where Type_local="Maison"			
AND Code_departement=71			

**4) Lister les noms des régions de France**

```
SELECT DISTINCT reg_ as nom_region  
FROM Region;
```



SQL 1	
1	SELECT DISTINCT reg_nom as nom_region
2	FROM Region
3	

	nom_region
1	Auvergne-Rhône-Alpes
2	Hauts-de-France
3	Provence-Alpes-Côte d'Azur
4	Grand Est
5	Occitanie
6	Normandie
7	Nouvelle-Aquitaine
8	Centre-Val de Loire
9	Corse
10	Bourgogne-Franche-Comté
11	Bretagne
12	Pays de la Loire
13	Île-de-France
14	Guadeloupe
15	Martinique
16	Guyane

##### 5) Le nombre de contrats sur les résidences principales

```
Select count (Type_contrat) as nbr_contrats_RP  
From Contrat  
Where Type_contrat= "Residence principale"
```



Structure de la Base de Données

Éditer les Pragma

Exécuter le SQL

SQL 1

```
1 Select count (Type_contrat) as nbr_contrats_RP
2 From Contrat
3 Where Type_contrat= "Residence principale"
4
```

	nbr_contrats_RP
1	25620

#### 6) La surface moyenne des logements avec un contrat à Paris

```
SELECT round ( avg (Surface) )  
FROM Contrat  
WHERE substr(code_postal, 1,2)="75"
```

SQL 1	
1	SELECT round ( avg (Surface))
2	FROM Contrat
3	WHERE substr(code_postal, 1,2)="75"
4	
round ( avg (Surface))	
1	52.0

#### 7) Les 5 contrats qui ont la surface la plus élevée

```
SELECT Contrat_ID as contrat, Surface  
FROM Contrat  
ORDER by Surface DESC LIMIT 5 ;
```



SQL 1

```
1 SELECT Contrat_ID as contrat, Surface
2 FROM Contrat
3 ORDER by Surface DESC LIMIT 5 ;
4
```

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

L'exécution s'est terminée sans erreur.  
Résultat : 5 enregistrements ramenés en 75ms  
À la ligne 1 :  
SELECT Contrat\_ID as contrat, Surface  
FROM Contrat  
ORDER by Surface DESC LIMIT 5 ;

#### 8) Prix moyen de la cotisation mensuelle

```
SELECT round (avg(Prix_cotisation_mensuel) )  
  
FROM Contrat ;
```



Structure de la Base de Données    Éditer les Pragmas    Exécuter le SQL

SQL 1

```
1 SELECT round (avg(Prix_cotisation_mensuel) )
2 FROM Contrat ;
3
```

	round (avg(Prix_cotisation_mensuel) )
1	19.0

L'exécution s'est terminée sans erreur.  
Résultat : 1 enregistrements ramenés en 68ms  
À la ligne 1 :  
SELECT round (avg(Prix\_cotisation\_mensuel) )  
FROM Contrat ;

9) **Nombre de contrats pour chaque catégorie déclarée des biens**

```
SELECT count (Contrat_ID) as contrats, Valeur_declaree_biens
FROM Contrat
GROUP by Valeur_declaree_biens ;
```



SQL 1

```
1 SELECT count (Contrat_ID) as contrats, Valeur_declaree_biens
2 FROM Contrat
3 GROUP by Valeur_declaree_biens ;
4
```

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

L'exécution s'est terminée sans erreur.  
Résultat : 4 enregistrements ramenés en 128ms  
À la ligne 1 :  
SELECT count (Contrat\_ID) as contrats, Valeur\_declaree\_biens  
FROM Contrat  
GROUP by Valeur\_declaree\_biens ;

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

```
SELECT dep_nom_num as departement, round (avg(Prix_cotisation_mensuel) ) as  
prix_moyen_cotisation  
  
FROM Region, Contrat  
  
WHERE region.code_postal=Contrat.Code_postal  
  
GROUP BY dep_nom_num  
  
ORDER by avg(Prix_cotisation_mensuel) DESC LIMIT 10 ;
```





SQL 1		
3	WHERE region.code_postal=Contrat.Code_postal	
4	GROUP BY dep_nom_num	
5	ORDER by avg(Prix_cotisation_mensuel) DESC LIMIT 10 ;	
6		
	departement	prix_moyen_cotisation
1	Paris (75)	36.0
2	Hauts-de-Seine (92)	26.0
3	Val-de-Marne (94)	20.0
4	Yvelines (78)	19.0
5	Rhône (69)	18.0
6	Ain (01)	18.0
7	Alpes-Maritimes (06)	18.0
8	Charente-Maritime (17)	17.0
9	Haute-Savoie (74)	17.0
10	Corse-du-Sud (2A)	17.0

11) Quel est le nombre de contrats avec formule Intégral pour la région Pays de la Loire

```
SELECT count(Contrat_ID),reg_nom
FROM Region, Contrat
WHERE Contrat.Code_postal=Region.code_postal
AND Formule="Integral" AND reg_nom="Pays de la Loire" ;
```



Structure de la Base de Données		Éditer les Pragmas	Exécuter le SQL	Parcourir
SQL 1				
2	FROM Region, Contrat			
3	WHERE Contrat.Code_postal=Region.code_postal			
4	AND Formule="Integral" AND reg_nom="Pays de la Loire" ;			
5				
	count(Contrat_ID)	reg_nom		
1	590	Pays de la Loire		

## 12) Lister les communes ayant eu au moins 150 contrats

SELECT count ( contrat ID), Commune

FROM Contrat

GROUP BY Commune

HAVING count (Contrat\_ID)>=150

Order by count ( contrat ID) desc



```
1 SELECT count (Contrat_ID), Commune
2 FROM Contrat
3 GROUP by Commune
4 HAVING count (Contrat_ID) >= 150
5 ORDER by count (Contrat_ID) DESC
```

	count (Contrat_ID)	Commune
1	515	PARIS 18
2	468	PARIS 17
3	407	PARIS 15
4	394	PARIS 16
5	387	NICE
6	381	PARIS 11
7	302	PARIS 20
8	302	BORDEAUX
9	291	NANTES
10	266	PARIS 19
11	263	PARIS 10

```
L'exécution s'est terminée sans erreur.
Résultat : 20 enregistrements ramenés en 149ms
À la ligne 1 :
SELECT count (Contrat_ID), Commune
FROM Contrat
GROUP by Commune
HAVING count (Contrat_ID) >= 150
```

### 13) Le nombre de contrats pour chaque région

```
SELECT count (Contrat_ID) as nbr_contrats, reg_nom as region
FROM Contrat, Region
WHERE Contrat.Code_postal=Region.code_postal
GROUP by reg_nom
Order by count (contrat ID) desc
```



SQL 1		
1	SELECT count ( Contrat_ID)as nbr_contrat, reg_nom as region	
2	FROM Contrat, Region	
3	WHERE Contrat.Code_postal=Region.code_postal	
4	GROUP BY reg_nom	
5	ORDER by count (Contrat_ID) DESC	
	nbr_contrat	region
1	14179	Île-de-France
2	3279	Provence-Alpes-Côte d'Azur
3	3056	Auvergne-Rhône-Alpes
4	2046	Nouvelle-Aquitaine
5	1601	Occitanie
6	1196	Pays de la Loire
7	1182	Hauts-de-France
8	947	Bretagne
9	822	Normandie
10	768	Grand Est
11	598	Centre-Val de Loire
12	287	Bourgogne-Franche-Comté
13	247	Corse
14	73	Martinique
15	37	Guyane

### Recherche des typologies de données

Pour chercher les typologies de données sur DB Browser, on ouvre la table dont on veut connaître les typologies et on clique sur parcourir la base de données, la table s'ouvre et sur la partie inférieure, on a les colonnes qui s'affichent avec le type des données. Ces typologies permettent à SQL de comprendre le type de donnée recherché.

### Utilisation des fonctions d'agrégation de données

Les notions de dimension ainsi que les fonctions d'agrégation de données m'ont été essentielles pour les requêtes.

En effet, l'utilisation des agrégations comme Distinct pour ne pas prendre en compte les doublons, Count pour compter le nombre d'enregistrements d'une table, AVG pour calculer la moyenne des données de la table et DESC pour les valeurs les plus grandes aux plus petites, m'a permis d'avoir des résultats plus précis des enregistrements de la base de données. Lors de la réalisation des requêtes.

