



SUPPORT DE PRÉSENTATION DU PROJET 3: REQUÊTEZ UNE BASE DE DONNÉES AVEC SQL

Présenté par : Aïda Gassama

1ère étape: Création du Dictionnaire des données

	Nom des colonnes	Type de données	Taille	Clé	Description
CONTRAT.CSV	Contrat_ID	INTEGER		Clé primaire	Id unique pour les contrats
	No_voie	INTEGER			Numéro dans la voie pour l'adresse du logement assuré
	B_T_Q	VARCHAR	1		Indicateur éventuel de répétition pour l'adresse du logement assuré sur un caractère
	Type_de_voie	VARCHAR	15		Type de voie pour l'adresse du logement assuré: rue, av (Avenue), rte (Route), ...
	Voie	VARCHAR	50		Libellé de la voie pour l'adresse du logement assuré
	Code_dep_code_commune	VARCHAR	5	Clé secondaire	Concaténation du code département et code commune pour avoir une clé unique
	Code_postal	VARCHAR	10		Code postal pour l'adresse du logement assuré
	Surface	INTEGER			Superficie du logement en m²
	Type_local	VARCHAR	15		Type de logement
	Occupation	VARCHAR	15		Statut d'occupation du logement de l'utilisateur : locataire ou propriétaire
	Type_contrat	VARCHAR	50		Type de contrat
	Formule	VARCHAR	15		Type de formule ou d'offre choisie
	Valeur_declaree_biens	VARCHAR	50		Plage de valeur déclarée des biens en euros, indiquée par des intervalles
	Prix_cotisation_mensuel	INTEGER			Montant mensuel de la cotisation en euros
REGION.CSV	Code_dep_code_commune	VARCHAR	15	Clé étrangère	Concaténation du code département et code commune pour avoir une clé unique
	reg_code	VARCHAR	2		Code unique de 2 chiffres représentant l'identifiant d'une région
	reg_nom	VARCHAR	50		Nom de la région
	aca_nom	VARCHAR	50		Nom de la ville
	dep_nom	VARCHAR	50		Nom du département
	com_nom_maj_court	VARCHAR	50		Nom de la commune en majuscules abrégé ou simplifié
	dep_code	INTEGER			Code numérique du département, représenté par un entier
	dep_nom_num	VARCHAR	50		Nom du département suivi de son code entre parenthèses

2ème étape Le Schéma Relationnel

Contrat
Contrat_ID: INTEGER NOT NULL [PK]
No_voie: INTEGER NOT NULL
B_T_Q: VARCHAR NOT NULL
Type_de_voie: VARCHAR NOT NULL
Voie: VARCHAR NOT NULL
Code_dep_code_commune: VARCHAR NOT NULL [FK]
Code_postal: VARCHAR NOT NULL
Surface: VARCHAR NOT NULL
Type_local: VARCHAR NOT NULL
Occupation: VARCHAR NOT NULL
Type_contrat: VARCHAR NOT NULL
Formule: VARCHAR NOT NULL
Valeur_declaree_biens: VARCHAR NOT NULL
Prix_cotisation_mensuel: INTEGER NOT NULL



Region
Code_dep_code_commune: VARCHAR NOT NULL [PK]
reg_code: VARCHAR NOT NULL
reg_nom: VARCHAR NOT NULL
aca_nom: VARCHAR NOT NULL
dep_nom: VARCHAR NOT NULL
com_nom_maj_court: VARCHAR NOT NULL
dep_code: INTEGER NOT NULL
dep_nom_num: VARCHAR NOT NULL

3ème étape: Chargement de la base de données dans MYSQL

Query 1 region region region region_csv contrat_csv

1 • `SELECT * FROM client.region_csv;`

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows: |

Code_dep_code_commune	reg_code	reg_nom	aca_nom	dep_nom	com_nom_maj_court	dep_code	dep_nom_num
51207	44	Grand Est	Reims	Marne	DAMPIERRE SUR AUVE	51	Marne (51)
51208	44	Grand Est	Reims	Marne	DAMPIERRE SUR MOIVRE	51	Marne (51)
51209	44	Grand Est	Reims	Marne	DAUCOURT	51	Marne (51)
51210	44	Grand Est	Reims	Marne	DIZY	51	Marne (51)
51211	44	Grand Est	Reims	Marne	DOMMARTIN DAMPIERRE	51	Marne (51)
51212	44	Grand Est	Reims	Marne	DOMMARTIN LETTREE	51	Marne (51)
51213	44	Grand Est	Reims	Marne	DOMMARTIN SOUS HANS	51	Marne (51)
51214	44	Grand Est	Reims	Marne	DOMMARTIN VARIMONT	51	Marne (51)
51215	44	Grand Est	Reims	Marne	DOMPREMY	51	Marne (51)
51216	44	Grand Est	Reims	Marne	DONTRIEN	51	Marne (51)
51217	44	Grand Est	Reims	Marne	DORMANS	51	Marne (51)

region_csv 1 x

Output

Action Output

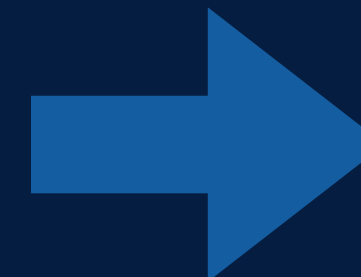
#	Time	Action	Message
✓ 1	00:34:01	SELECT * FROM client.contrat_csv	30335 row(s) returned
✓ 2	00:34:07	SELECT * FROM client.region_csv	38916 row(s) returned

Query 1 region region region region_csv contrat_csv contrat_csv

1 • `SELECT * FROM client.contrat_csv;`

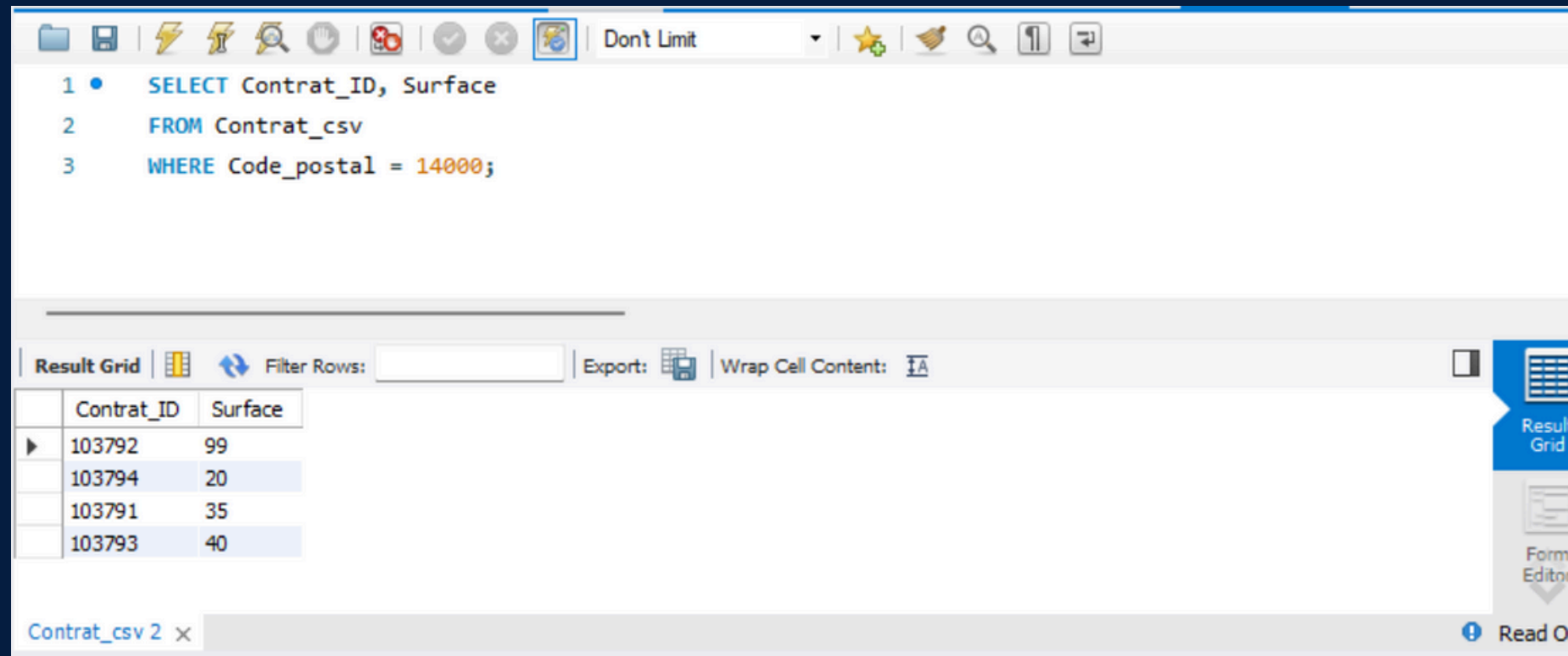
Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows: |

Contrat_ID	No_voie	B_T_Q	Type_de_voie	Voie	Code_dep_code_commune	Code_postal	Surface	Type_local	Occupation	Type_contrat	Formule	Valeur_dec
100602	347		RUE	DU CHATEAU	1103	1170	48	Appartement	Locataire	Residence principale	Classique	0-25000
100605	39		RUE	BUFFON	1033	1200	109	Appartement	Locataire	Residence principale	Classique	0-25000
100606	8		RUE	DE GENEVE	1354	1630	53	Appartement	Proprietaire	Residence principale	Classique	0-25000
100609	226		ALL	DES CAPUCINES	1354	1630	117	Maison	Proprietaire	Residence principale	Classique	25000-5000
100611	79		CRS	DE VERDUN	1283	1100	138	Appartement	Proprietaire	Residence secondaire	Classique	0-25000
100612	240		RUE	DE PRE BAILLY	1173	1170	45	Appartement	Locataire	Residence principale	Classique	0-25000
100613	3		RUE	TURENNE	1033	1200	83	Appartement	Locataire	Residence principale	Classique	0-25000
100615	59		RUE	ALEXANDRE BERARD	1004	1500	165	Appartement	Locataire	Residence principale	Classique	25000-5000
100616	282		CHE	DES LONGES RAYES	1071	1170	42	Appartement	Proprietaire	Residence principale	Classique	0-25000
100617	54		GR	GRANDE RUE	1396	1150	68	Appartement	Proprietaire	Residence secondaire	Classique	0-25000



- Création de la base de données puis chargement de la base de données à partir du fichier CSV sur MYSQL
- Création des 2 tables: Region_cvs et Contrat_cvs

4ème étape: Les requêtes sur MySQL



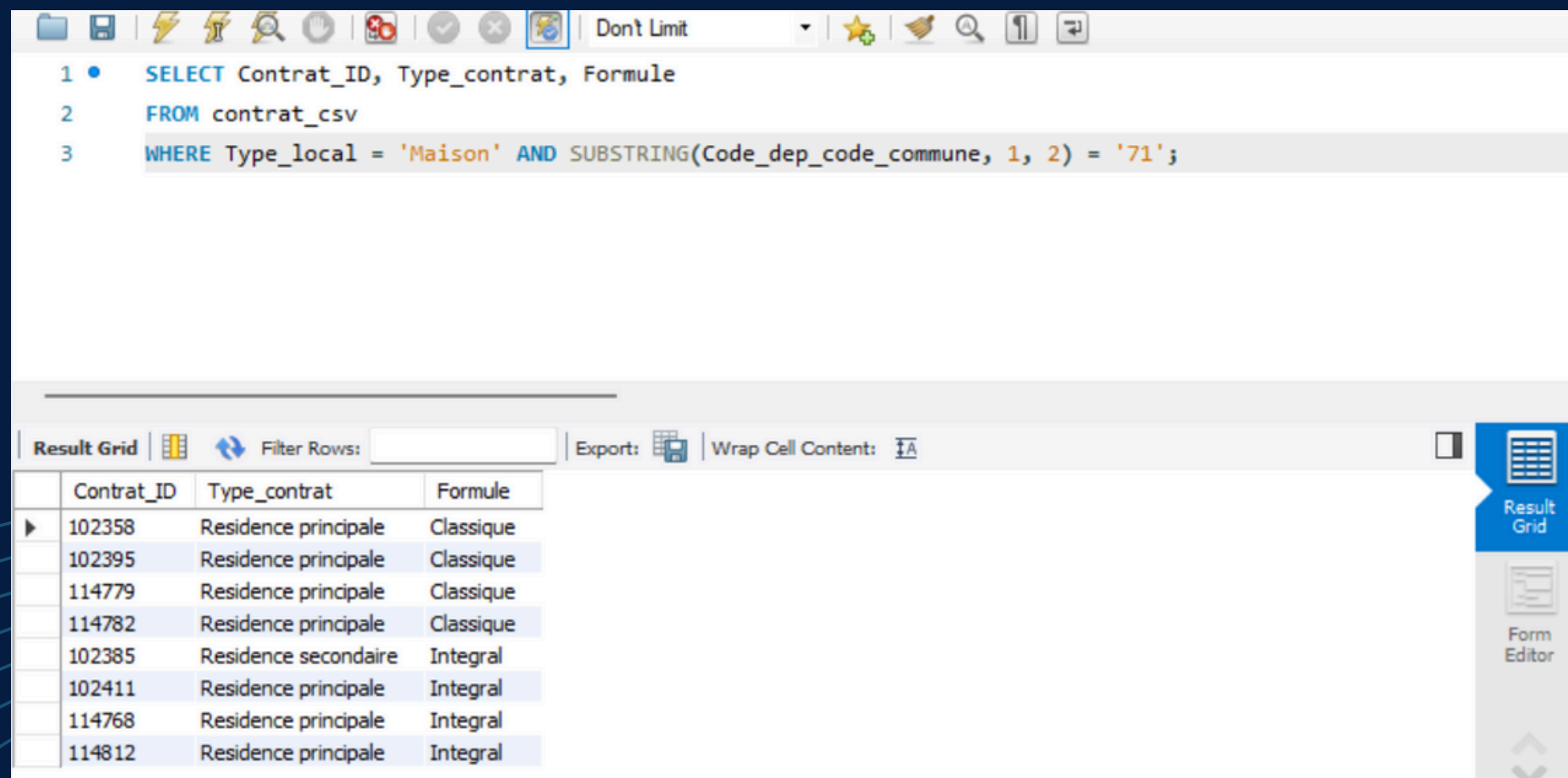
```
1 • SELECT Contrat_ID, Surface
2   FROM Contrat_csv
3  WHERE Code_postal = 14000;
```

Contrat_ID	Surface
103792	99
103794	20
103791	35
103793	40

Requête 1:

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

- **SELECT Contrat_ID, Surface** : La requête sélectionne deux colonnes de la table : Contrat_ID et Surface
- **FROM Contrat_csv** : Indique que les données doivent être extraites de la table Contrat_csv
- **WHERE Code_postal = 14000** : La requête filtre les résultats pour ne retourner que les lignes où la valeur de la colonne Code_postal est égale à 14000 (Caen)



```
1 • SELECT Contrat_ID, Type_contrat, Formule
2   FROM contrat_csv
3  WHERE Type_local = 'Maison' AND SUBSTRING(Code_dep_code_commune, 1, 2) = '71';
```

Contrat_ID	Type_contrat	Formule
102358	Residence principale	Classique
102395	Residence principale	Classique
114779	Residence principale	Classique
114782	Residence principale	Classique
102385	Residence secondaire	Integral
102411	Residence principale	Integral
114768	Residence principale	Integral
114812	Residence principale	Integral

Requête 2:

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

- **SELECT Contrat_ID, Type_contrat, Formula** : Permet de sélectionner 3 colonnes de la table contrat_csv
- **FROM contrat_csv** : Indique que les données doivent être extraites de la table contrat_csv.
- **WHERE Type_local = 'Maison'** : La requête filtre les résultats pour ne retourner que les lignes où Type_local est 'Maison'
- **AND SUBSTRING(Code_dep_code_commune, 1, 2) = '71'** : Cette partie de la condition filtre les résultats pour ne retourner que les lignes où les deux premiers caractères de la colonne Code_dep_code_commune sont '71'. Cela correspond au département 71 (Saône-et-Loire en France)

Les requêtes sur MySQL

```
1 • SELECT DISTINCT r.reg_nom
2 FROM contrat_csv c
3 JOIN region_csv r ON substring(c.Code_dep_code_commune,1, 2) = r.dep_code;
```

Result Grid

reg_nom
Bourgogne-Franche-Comté
Bretagne
Auvergne-Rhône-Alpes
Pays de la Loire
Hauts-de-France
Ile-de-France
Corse

Requête 3:

Lister le nom des régions de France

- **SELECT DISTINCT r.reg_nom** : La requête sélectionne la colonne reg_nom (nom de la région) de la table region_csv. DISTINCT garantit que chaque région n'apparaît qu'une seule fois dans les résultats, même si elle est associée à plusieurs contrats
- **FROM contrat_csv c** : La table principale utilisée est contrat_csv, à laquelle on donne l'alias c pour simplifier les références
- **JOIN region_csv r ON SUBSTRING (c.Code_dep_code_commune, 1, 2) : r.dep_code** : Une jointure est effectuée avec la table region_csv (r). Elle est basée sur les deux premiers caractères de la colonne Code_dep_code_commune dans contrat_csv, qui représentent le code départemental

```
1 • SELECT Contrat_ID, Surface
2 FROM contrat_csv
3 ORDER BY CAST(Surface AS DECIMAL) DESC
4 LIMIT 5;
```

Result Grid

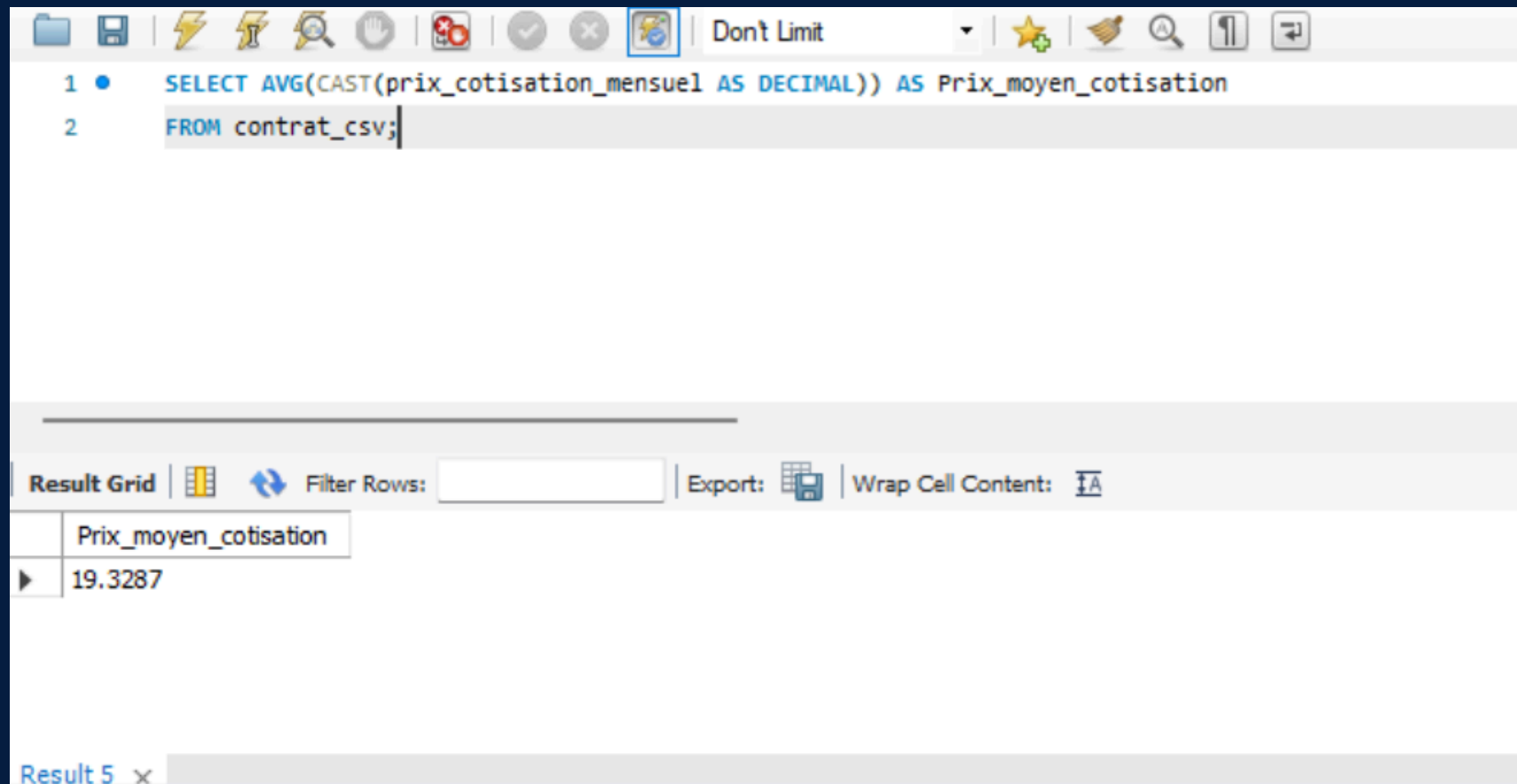
Contrat_ID	Surface
104211	815
105463	742
130878	595
100822	570
109872	559

Requête 4 :

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

- **SELECT Contrat_ID, Surface** : La requête sélectionne deux colonnes de la table : Contrat_ID et Surface
- **FROM Contrat_csv** : Indique que les données doivent être extraites de la table Contrat_csv
- **ORDER BY CAST(Surface AS DECIMAL) DESC** : ORDER BY trie les résultats en fonction de la colonne Surface
- **CAST(Surface AS DECIMAL)** convertit la colonne Surface en un type numérique (DECIMAL) pour s'assurer que le tri se fait correctement
- **DESC** indique que le tri est en ordre décroissant (du plus grand au plus petit)
- **LIMIT 5** : Restreint les résultats aux 5 premiers enregistrements après le tri

Les requêtes sur MySQL



The screenshot shows the MySQL Workbench interface. The SQL editor contains the following query:

```
1 • SELECT AVG(CAST(prix_cotisation_mensuel AS DECIMAL)) AS Prix_moyen_cotisation
2 FROM contrat_csv;
```

The 'Result Grid' tab is active, displaying the result of the query:

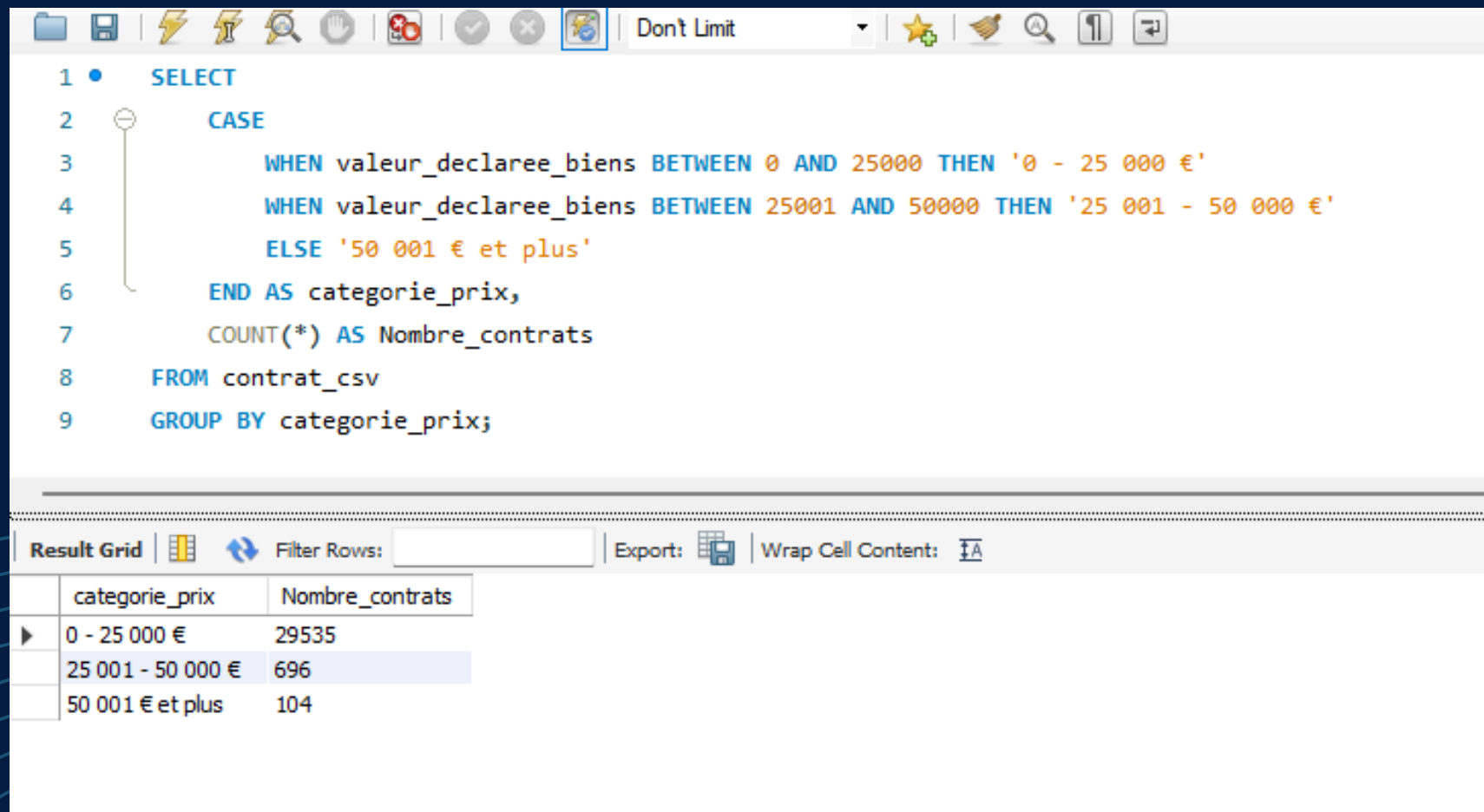
Prix_moyen_cotisation
19.3287

Result 5 x

Requête 5 :

Quel est le prix moyen de la cotisation mensuelle ?

- **SELECT** : permet de sélectionner des données
- **AVG** : est une fonction d'agrégation qui calcule la moyenne des valeurs dans une colonne
- **AS Prix_moyen_cotisation** : donne un alias à la colonne qui va s'afficher
- **FROM** : Spécifie la table à partir de laquelle les données sont sélectionnées. Dans ce cas, la table est nommée **contrat_csv**



The screenshot shows the MySQL Workbench interface. The SQL editor contains the following query:

```
1 • SELECT
2 CASE
3     WHEN valeur_declaree_biens BETWEEN 0 AND 25000 THEN '0 - 25 000 €'
4     WHEN valeur_declaree_biens BETWEEN 25001 AND 50000 THEN '25 001 - 50 000 €'
5     ELSE '50 001 € et plus'
6 END AS categorie_prix,
7 COUNT(*) AS Nombre_contrats
8 FROM contrat_csv
9 GROUP BY categorie_prix;
```

The 'Result Grid' tab is active, displaying the result of the query:

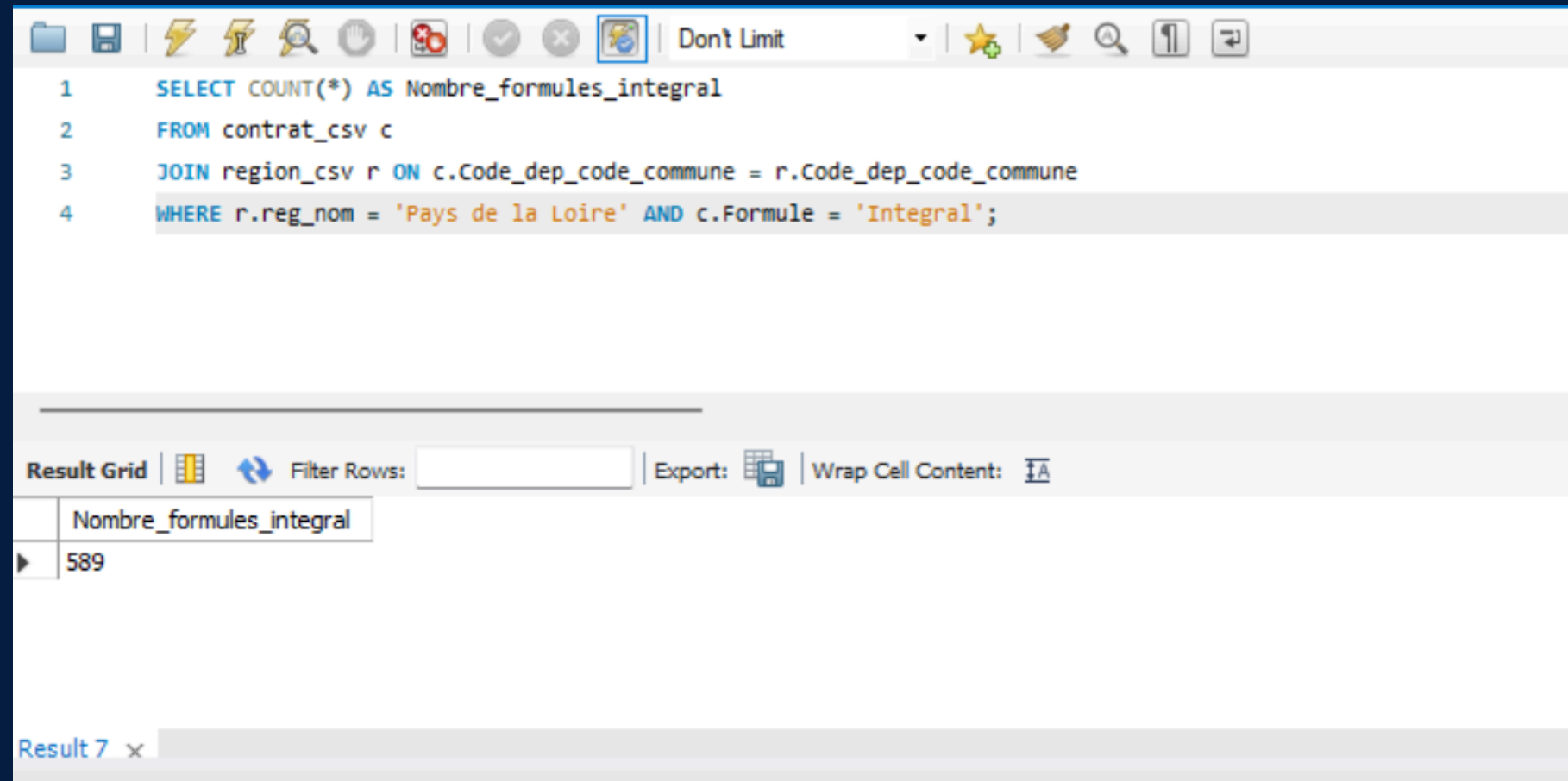
categorie_prix	Nombre_contrats
0 - 25 000 €	29535
25 001 - 50 000 €	696
50 001 € et plus	104

Requête 6 :

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

- **CASE** : est une expression conditionnelle en SQL qui permet de créer des branches logiques
- **WHEN valeur_declaree_biens BETWEEN 0 AND 25000 THEN '0 - 25 000 €'** : Si la valeur de `valeur_declaree_biens` est comprise entre 0 et 25 000, alors elle est catégorisée comme '0 - 25 000 €'
- **ELSE '50 001 € et plus'** : Si la valeur de `valeur_declaree_biens` ne correspond à aucune des plages précédentes, elle est catégorisée comme '50 001 € et plus'.
- **COUNT(*)** compte le nombre total de lignes (contrats) dans chaque catégorie
- **GROUP BY** est utilisé pour regrouper les résultats par la colonne

Les requêtes sur MySQL



The screenshot shows the MySQL Workbench interface. The SQL editor contains the following query:

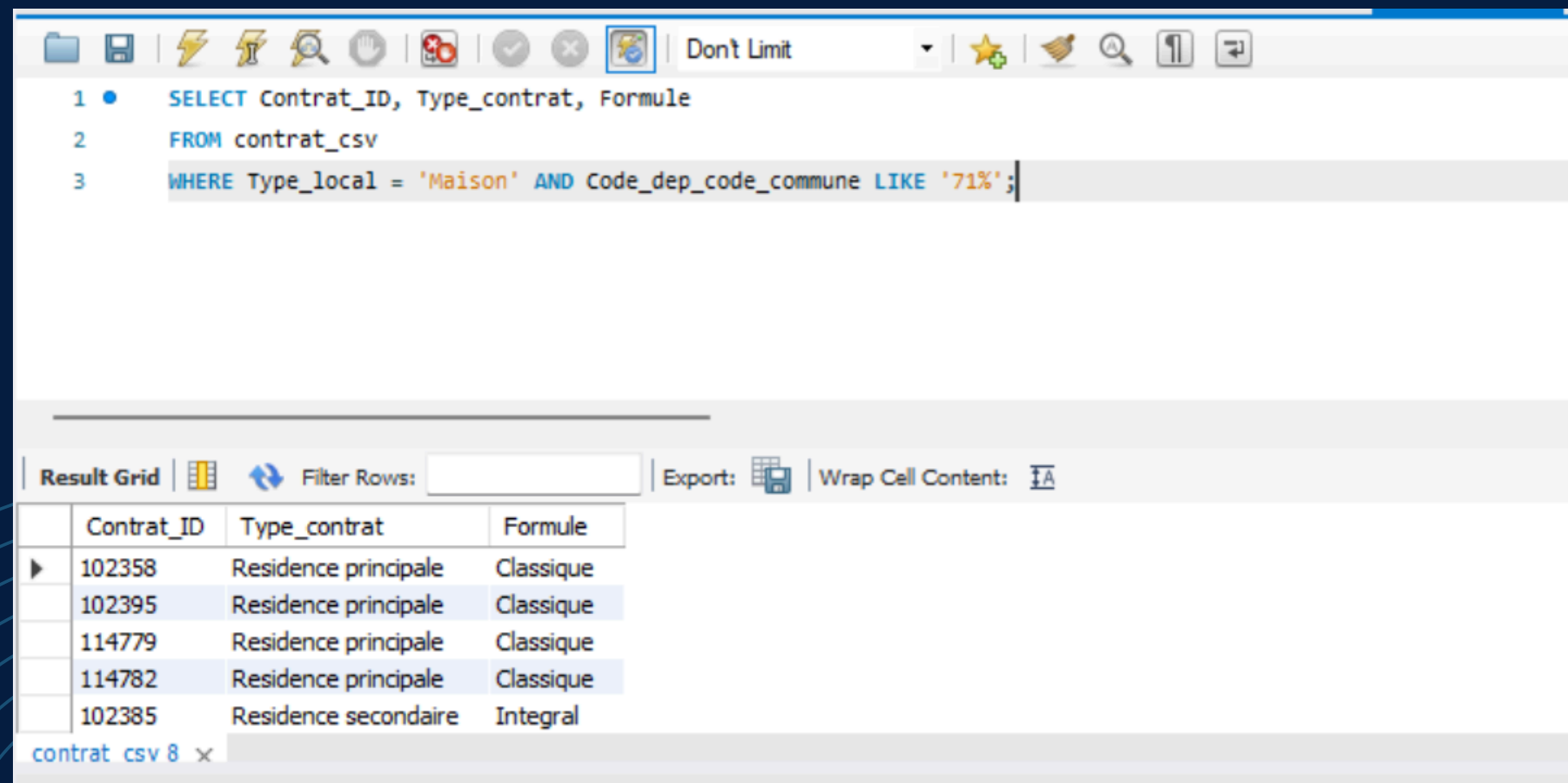
```
1 SELECT COUNT(*) AS Nombre_formules_integral
2 FROM contrat_csv c
3 JOIN region_csv r ON c.Code_dep_code_commune = r.Code_dep_code_commune
4 WHERE r.reg_nom = 'Pays de la Loire' AND c.Formule = 'Integral';
```

The result grid shows a single row with the value 589 for the column Nombre_formules_integral.

Nombre_formules_integral
589

Requête 7 :

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



The screenshot shows the MySQL Workbench interface. The SQL editor contains the following query:

```
1 SELECT Contrat_ID, Type_contrat, Formule
2 FROM contrat_csv
3 WHERE Type_local = 'Maison' AND Code_dep_code_commune LIKE '71%';
```

The result grid shows a table with 5 rows and 3 columns: Contrat_ID, Type_contrat, and Formule.

Contrat_ID	Type_contrat	Formule
102358	Residence principale	Classique
102395	Residence principale	Classique
114779	Residence principale	Classique
114782	Residence principale	Classique
102385	Residence secondaire	Integral

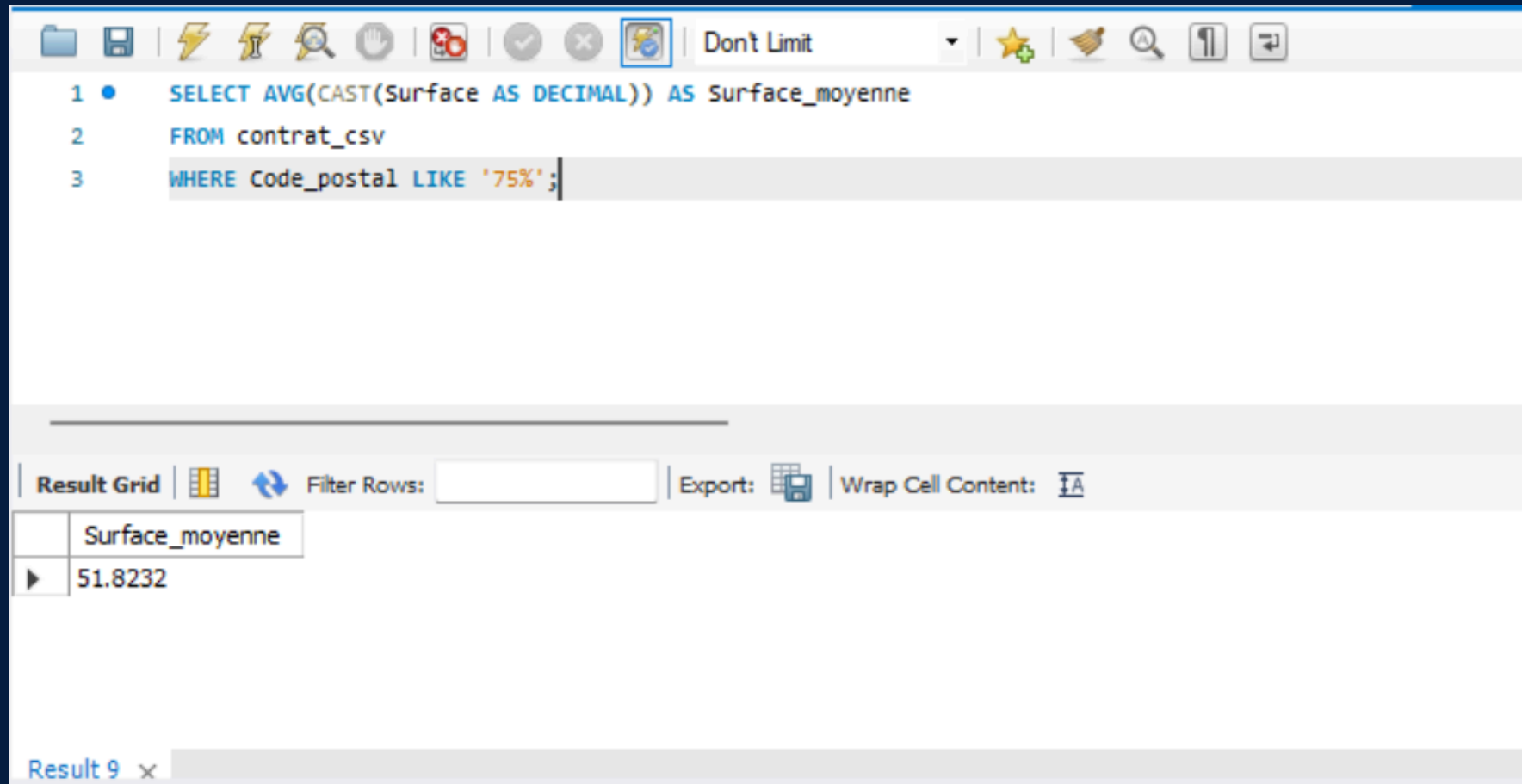
Requête 8 :

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

Les requêtes sur MySQL

Requête 9 :

Quelle est la surface moyenne des contrats à Paris ?



```
1 • SELECT AVG(CAST(Surface AS DECIMAL)) AS Surface_moyenne
2 FROM contrat_csv
3 WHERE Code_postal LIKE '75%';
```

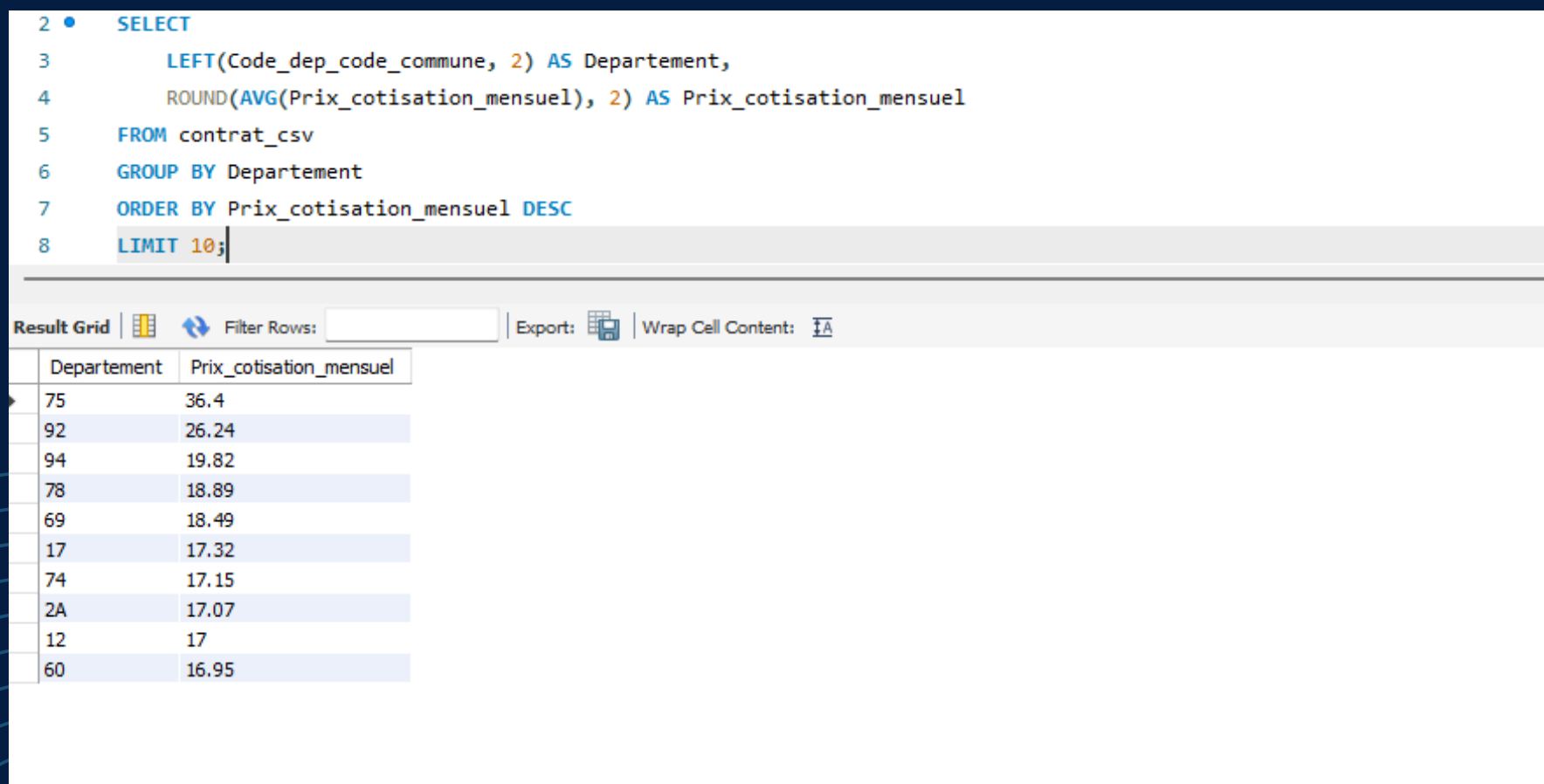
Result Grid

Surface_moyenne
51.8232

Result 9 x

Requête 10 :

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

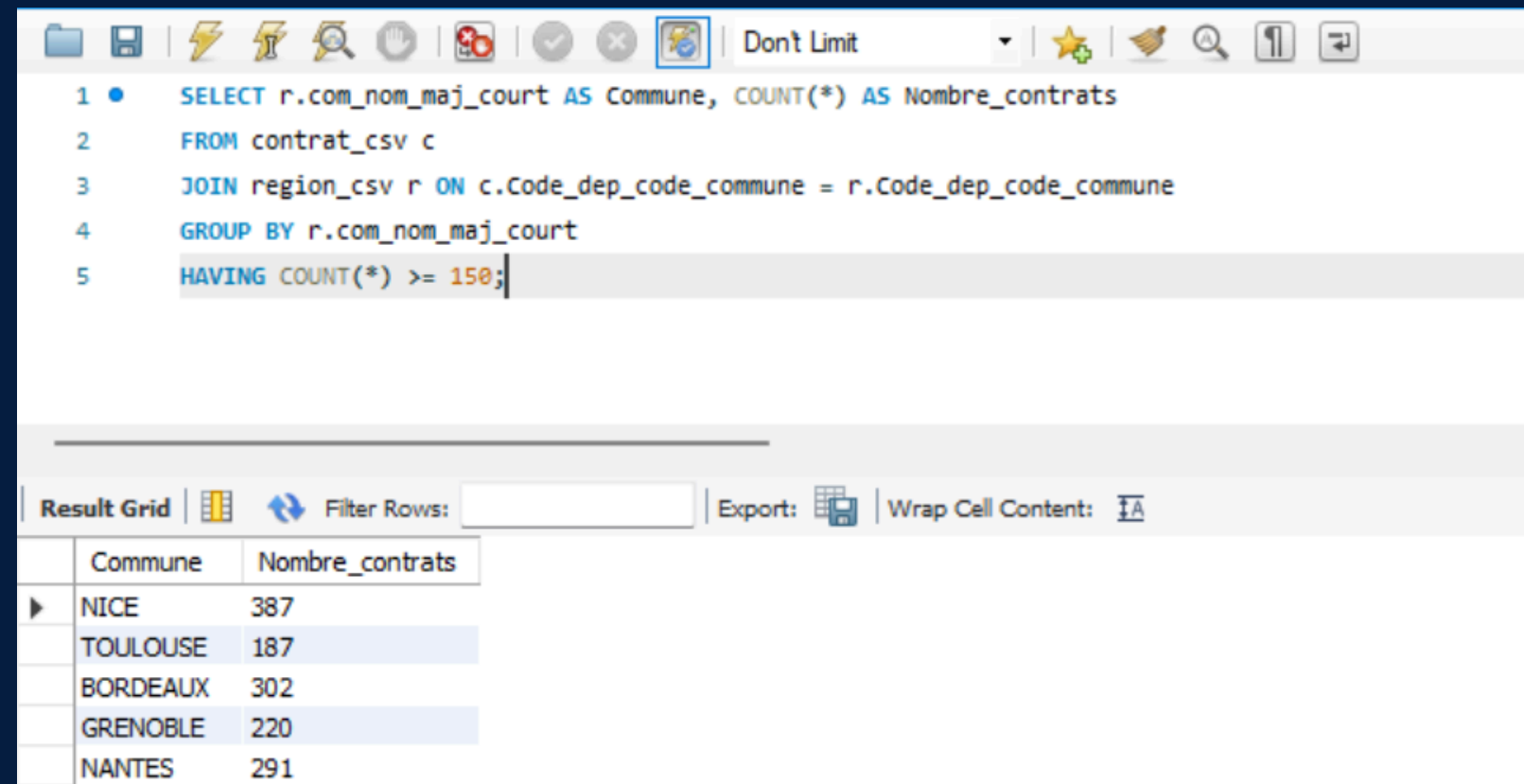


```
2 • SELECT
3     LEFT(Code_dep_code_commune, 2) AS Departement,
4     ROUND(AVG(Prix_cotisation_mensuel), 2) AS Prix_cotisation_mensuel
5 FROM contrat_csv
6 GROUP BY Departement
7 ORDER BY Prix_cotisation_mensuel DESC
8 LIMIT 10;
```

Result Grid

Departement	Prix_cotisation_mensuel
75	36.4
92	26.24
94	19.82
78	18.89
69	18.49
17	17.32
74	17.15
2A	17.07
12	17
60	16.95

Les requêtes sur MySQL



The screenshot shows a MySQL Workbench window with a SQL query editor and a result grid. The query is as follows:

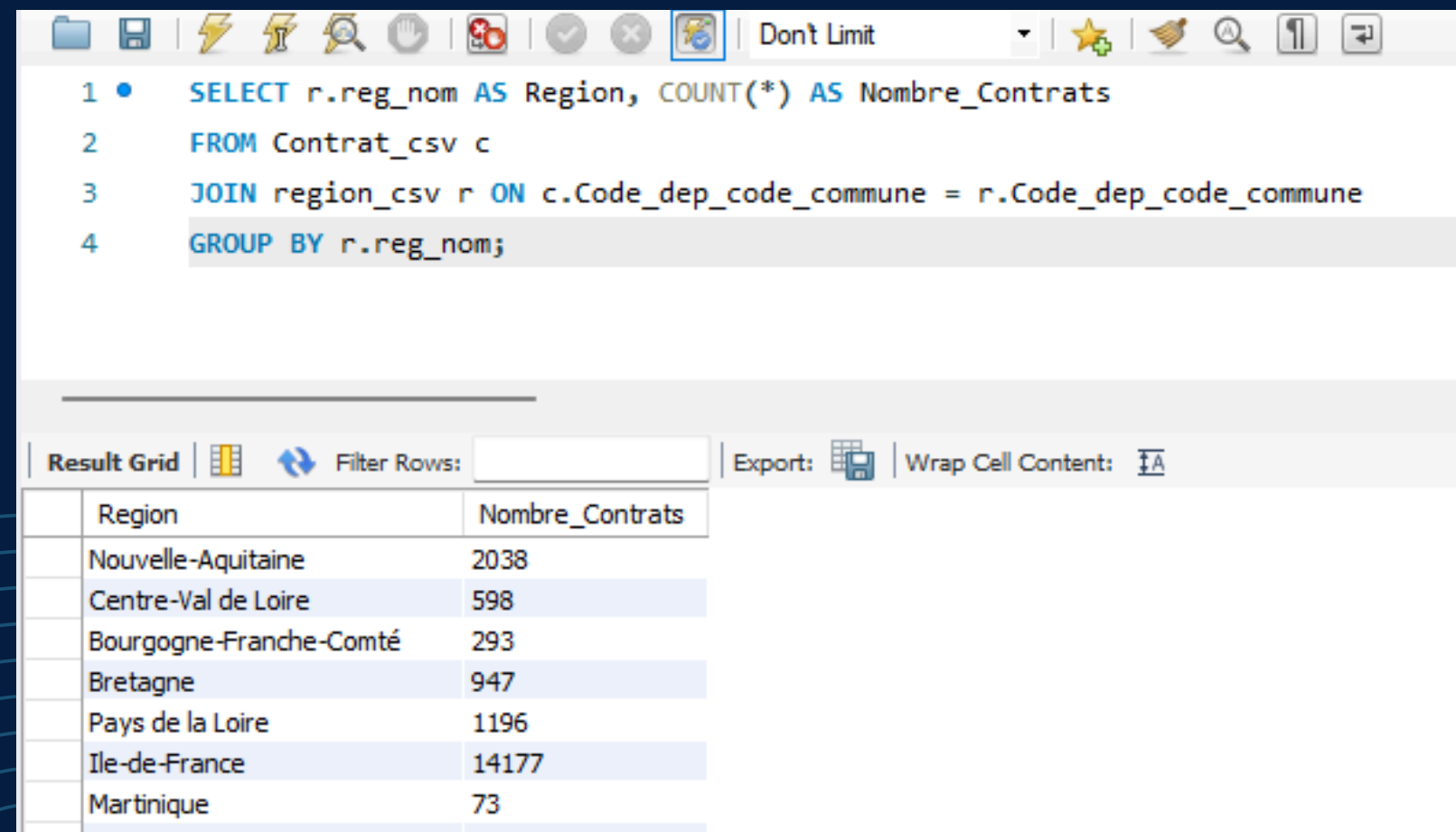
```
1 • SELECT r.com_nom_maj_court AS Commune, COUNT(*) AS Nombre_contrats
2 FROM contrat_csv c
3 JOIN region_csv r ON c.Code_dep_code_commune = r.Code_dep_code_commune
4 GROUP BY r.com_nom_maj_court
5 HAVING COUNT(*) >= 150;
```

The result grid displays the following data:

Commune	Nombre_contrats
NICE	387
TOULOUSE	187
BORDEAUX	302
GRENOBLE	220
NANTES	291

Requête 11 :

Liste des communes ayant eu au moins 150 contrats



The screenshot shows a MySQL Workbench window with a SQL query editor and a result grid. The query is as follows:

```
1 • SELECT r.reg_nom AS Region, COUNT(*) AS Nombre_Contrats
2 FROM Contrat_csv c
3 JOIN region_csv r ON c.Code_dep_code_commune = r.Code_dep_code_commune
4 GROUP BY r.reg_nom;
```

The result grid displays the following data:

Region	Nombre_Contrats
Nouvelle-Aquitaine	2038
Centre-Val de Loire	598
Bourgogne-Franche-Comté	293
Bretagne	947
Pays de la Loire	1196
Ile-de-France	14177
Martinique	73

Requête 12 :

Quel est le nombre de contrats pour chaque région ?