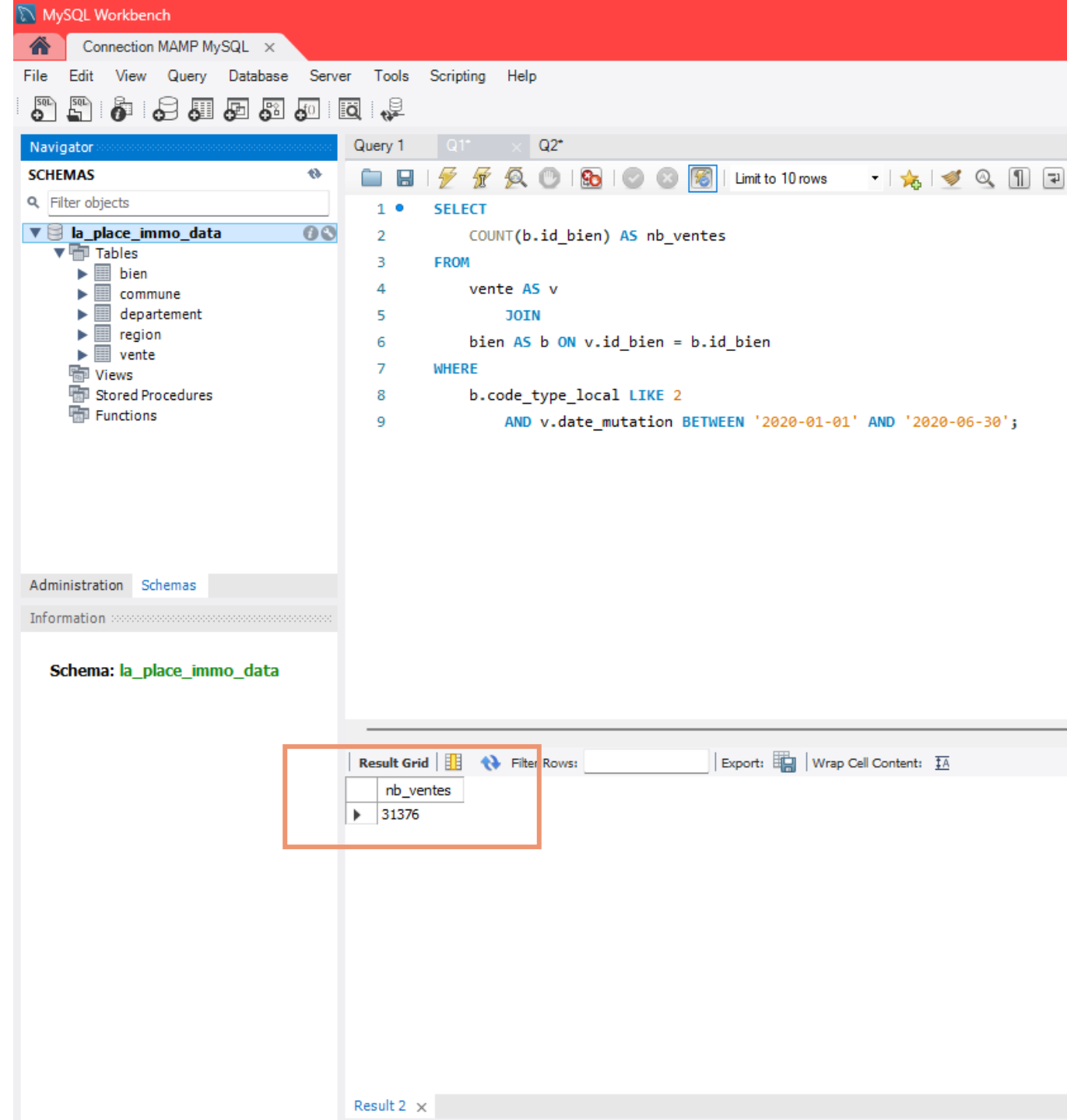


NOMBRE TOTAL D'APPARTEMENTS VENDUS AU 1ER SEMESTRE 2020.

```
SELECT COUNT(b.id_bien) AS nb_ventes  
FROM vente AS v  
JOIN bien AS b ON v.id_bien = b.id_bien  
WHERE b.code_type_local LIKE 2  
AND v.date_mutation BETWEEN '2020-01-  
01' AND '2020-06-30';
```



The screenshot shows the MySQL Workbench interface. The top menu bar includes File, Edit, View, Query, Database, Server, Tools, Scripting, and Help. The left sidebar shows the Navigator pane with the SCHEMAS section expanded, displaying the database structure for 'la_place_immo_data'. The main query editor shows the following SQL query:

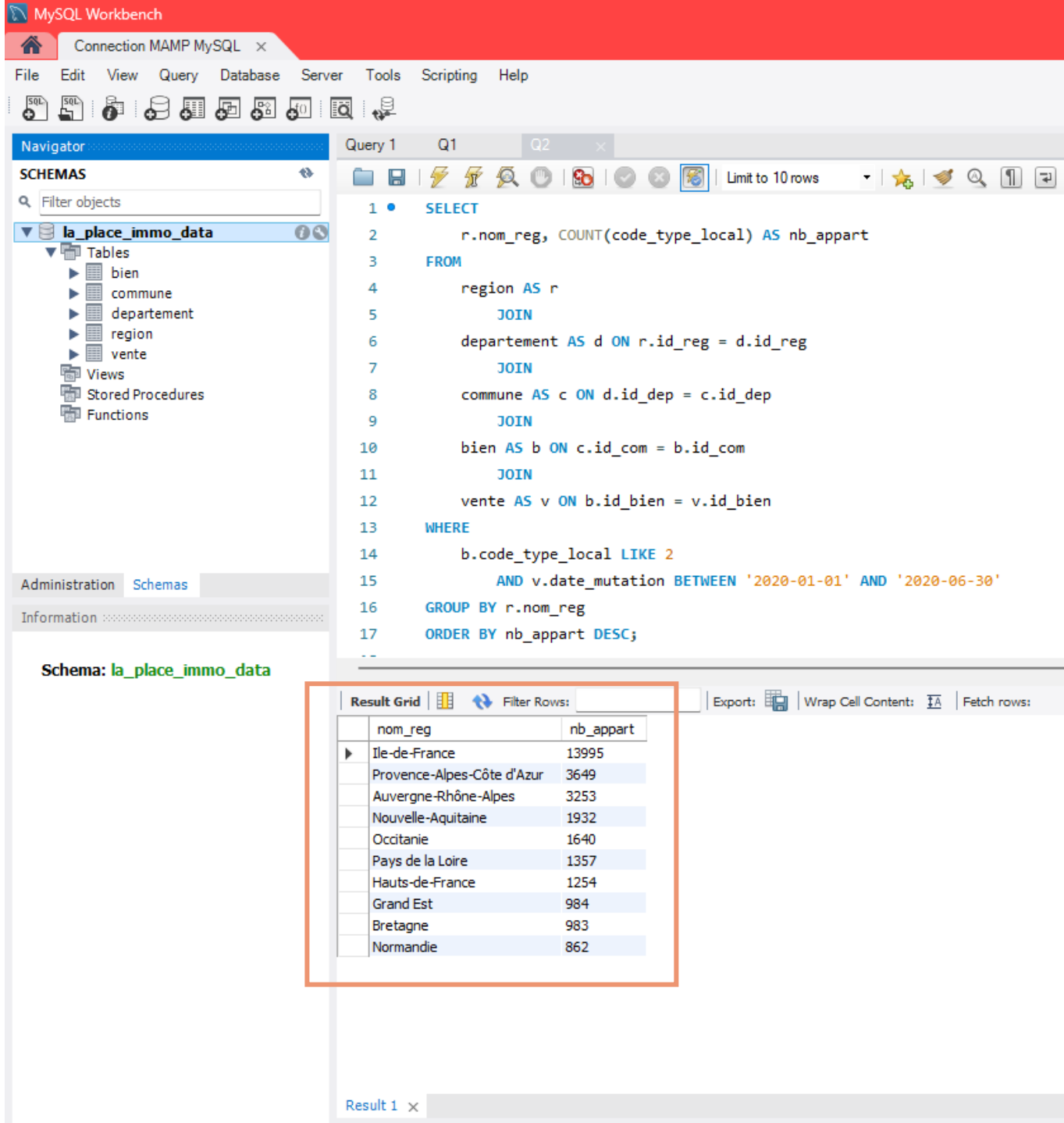
```
1 • SELECT  
2     COUNT(b.id_bien) AS nb_ventes  
3 FROM  
4     vente AS v  
5     JOIN  
6     bien AS b ON v.id_bien = b.id_bien  
7 WHERE  
8     b.code_type_local LIKE 2  
9     AND v.date_mutation BETWEEN '2020-01-01' AND '2020-06-30';
```

The bottom right pane shows the result grid for 'Result 2'. The result is a single row with the value 31376 for the column nb_ventes.

nb_ventes
31376

NOMBRE DE VENTES D'APPARTEMENT PAR RÉGION POUR LE 1ER SEMESTRE 2020.

```
SELECT r.nom_reg, COUNT(code_type_local) AS
nb_appart FROM region AS r
JOIN departement AS d ON r.id_reg = d.id_reg
JOIN commune AS c ON d.id_dep = c.id_dep
JOIN bien AS b ON c.id_com = b.id_com
JOIN vente AS v ON b.id_bien = v.id_bien
WHERE b.code_type_local LIKE 2 AND
v.date_mutation BETWEEN '2020-01-01' AND
'2020-06-30' GROUP BY r.nom_reg
ORDER BY nb_appart DESC;
```

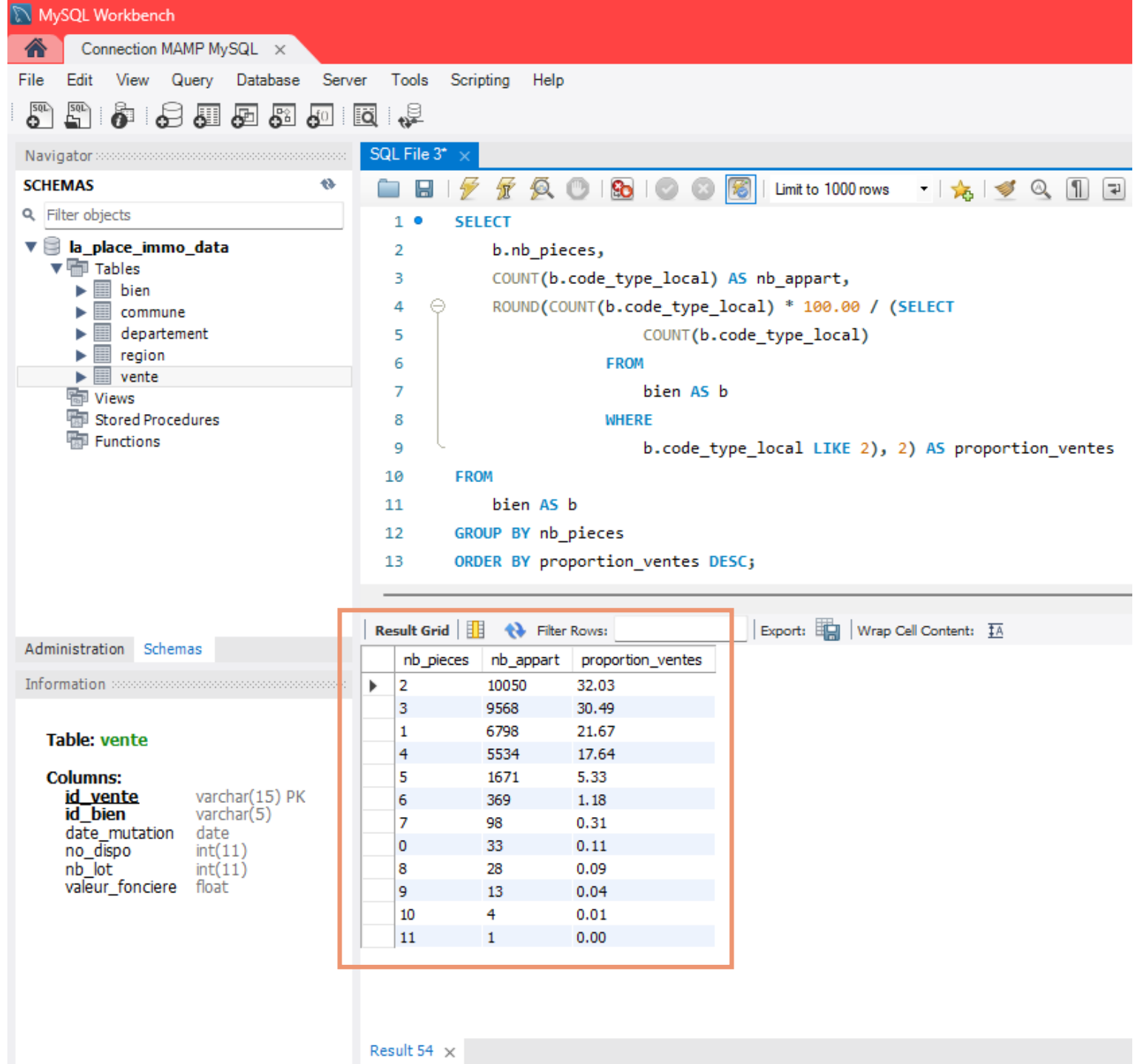


The screenshot shows the MySQL Workbench interface. The top menu bar includes File, Edit, View, Query, Database, Server, Tools, Scripting, and Help. The left sidebar shows the 'Navigator' pane with 'SCHEMAS' and a search filter. The 'la_place_immo_data' database is selected, showing a tree of tables (bien, commune, departement, region, vente), Views, Stored Procedures, and Functions. The 'Query 1' pane displays the SQL query from the previous block. The 'Result Grid' pane shows the results of the query, which are sorted by 'nb_appart' in descending order.

nom_reg	nb_appart
Ile-de-France	13995
Provence-Alpes-Côte d'Azur	3649
Auvergne-Rhône-Alpes	3253
Nouvelle-Aquitaine	1932
Occitanie	1640
Pays de la Loire	1357
Hauts-de-France	1254
Grand Est	984
Bretagne	983
Normandie	862

PROPORTION DES VENTES D'APPARTEMENTS PAR LE NOMBRE DE PIÈCES.

```
SELECT b.nb_pieces,
COUNT(b.code_type_local) AS nb_appart,
ROUND(COUNT(b.code_type_local) * 100.00
/ (SELECT
COUNT(b.code_type_local) FROM
bien AS b WHERE
b.code_type_local LIKE 2), 2) AS
proportion_ventes FROM bien AS b
GROUP BY nb_pieces
ORDER BY proportion_ventes DESC;
```



The screenshot shows the MySQL Workbench interface. The top menu bar includes File, Edit, View, Query, Database, Server, Tools, Scripting, and Help. The left sidebar shows the 'Schemas' panel with a tree view of the 'la_place_immo_data' database, including tables (bien, commune, departement, region, vente), views, stored procedures, and functions. The 'vente' table is selected.

The main editor displays a SQL query in a file named 'SQL File 3*':

```
1 SELECT
2     b.nb_pieces,
3     COUNT(b.code_type_local) AS nb_appart,
4     ROUND(COUNT(b.code_type_local) * 100.00 / (SELECT
5         COUNT(b.code_type_local)
6     FROM
7         bien AS b
8     WHERE
9         b.code_type_local LIKE 2), 2) AS proportion_ventes
10 FROM
11     bien AS b
12 GROUP BY nb_pieces
13 ORDER BY proportion_ventes DESC;
```

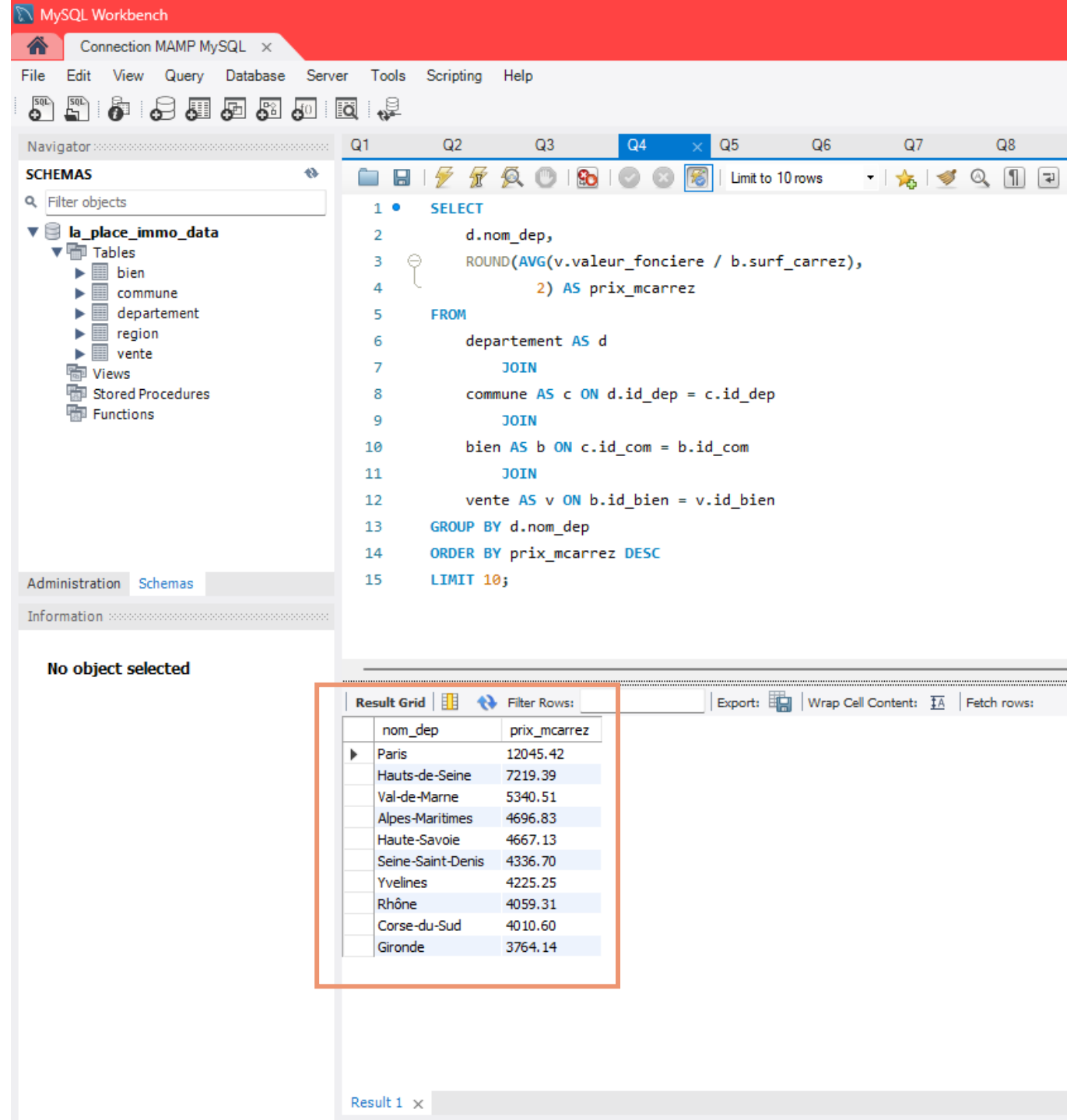
The 'Result Grid' at the bottom shows the query results, sorted by 'proportion_ventes' in descending order. The results are as follows:

	nb_pieces	nb_appart	proportion_ventes
2	10050	32.03	
3	9568	30.49	
1	6798	21.67	
4	5534	17.64	
5	1671	5.33	
6	369	1.18	
7	98	0.31	
0	33	0.11	
8	28	0.09	
9	13	0.04	
10	4	0.01	
11	1	0.00	

The 'Information' panel at the bottom left shows details for the 'vente' table, including columns: id_vente (varchar(15) PK), id_bien (varchar(5)), date_mutation (date), no_dispo (int(11)), nb_lot (int(11)), and valeur_fonciere (float).

LISTE DES 10 DÉPARTEMENTS
OÙ LE PRIX DU MÈTRE CARRÉ EST LE PLUS ÉLEVÉ.

```
SELECT d.nom_dep,
ROUND(AVG(v.valeur_fonciere / b.surf_carrez), 2)
AS prix_mcarrez FROM departement AS d
JOIN commune AS c ON d.id_dep = c.id_dep
JOIN bien AS b ON c.id_com = b.id_com
JOIN vente AS v ON b.id_bien = v.id_bien
GROUP BY d.nom_dep
ORDER BY prix_mcarrez DESC
LIMIT 10;
```



The screenshot shows the MySQL Workbench interface. The 'Schemas' pane on the left displays the database structure for 'la_place_immo_data', including tables like 'bien', 'commune', 'departement', 'region', and 'vente'. The 'Query' pane on the right contains the SQL query used to find the top 10 departments by average price per square meter. The 'Result Grid' at the bottom right shows the output of the query, listing the department names and their corresponding average price per square meter.

SQL Query:

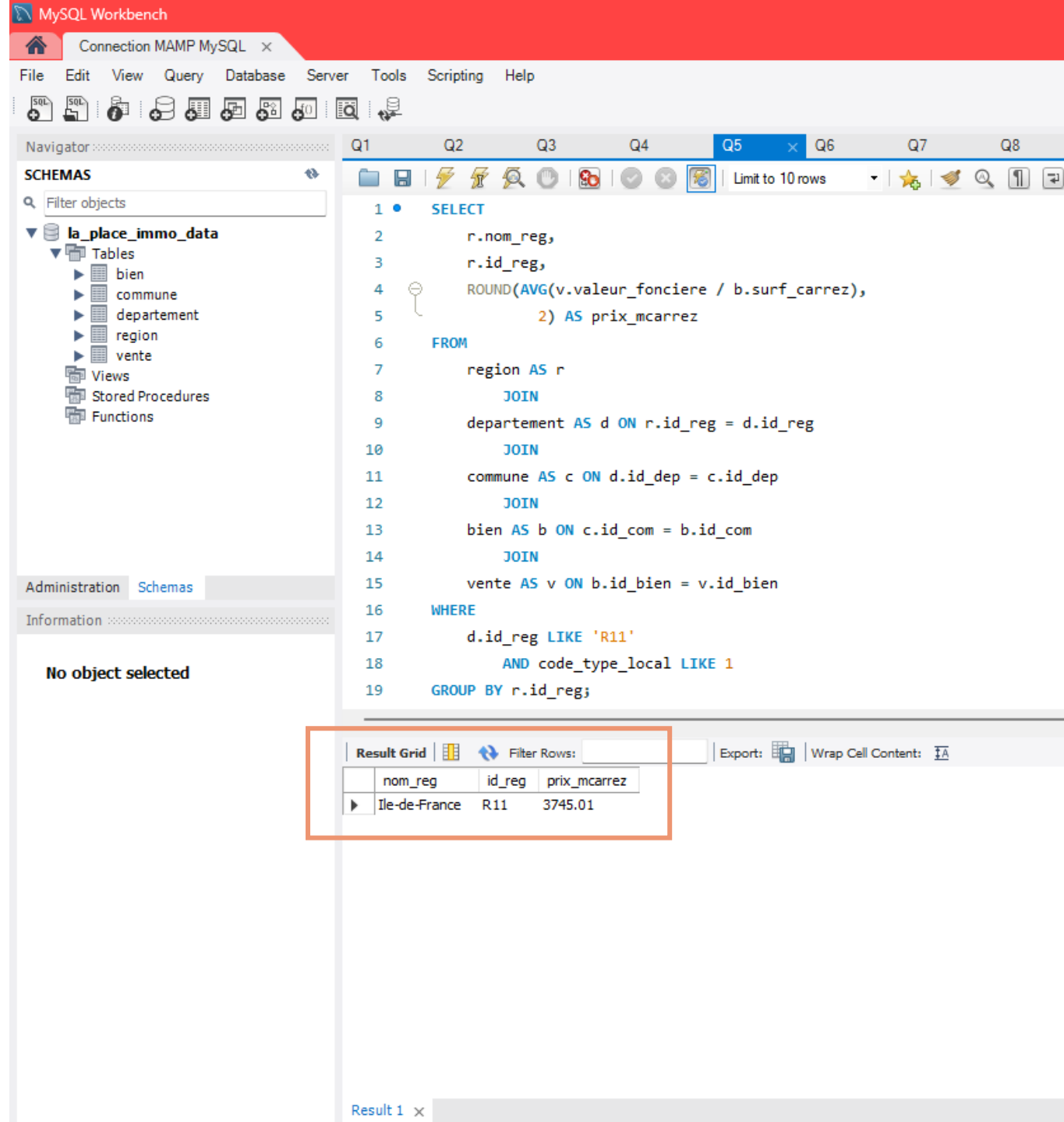
```
SELECT
  d.nom_dep,
  ROUND(AVG(v.valeur_fonciere / b.surf_carrez), 2) AS prix_mcarrez
FROM
  departement AS d
  JOIN
    commune AS c ON d.id_dep = c.id_dep
  JOIN
    bien AS b ON c.id_com = b.id_com
  JOIN
    vente AS v ON b.id_bien = v.id_bien
GROUP BY d.nom_dep
ORDER BY prix_mcarrez DESC
LIMIT 10;
```

Result Grid:

nom_dep	prix_mcarrez
Paris	12045.42
Hauts-de-Seine	7219.39
Val-de-Marne	5340.51
Alpes-Maritimes	4696.83
Haute-Savoie	4667.13
Seine-Saint-Denis	4336.70
Yvelines	4225.25
Rhône	4059.31
Corse-du-Sud	4010.60
Gironde	3764.14

PRIX MOYEN DU MÈTRE CARRÉ D'UNE MAISON EN ÎLE-DE-FRANCE.

```
SELECT r.nom_reg, r.id_reg,
ROUND(AVG(v.valeur_fonciere / b.surf_carrez), 2)
AS prix_mcarrez FROM region AS r
JOIN departement AS d ON r.id_reg = d.id_reg
JOIN commune AS c ON d.id_dep = c.id_dep
JOIN bien AS b ON c.id_com = b.id_com
JOIN vente AS v ON b.id_bien = v.id_bien
WHERE d.id_reg LIKE 'R11'
AND code_type_local LIKE 1
GROUP BY r.id_reg;
```



The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' tree for the 'la_place_immo_data' database, listing tables: bien, commune, departement, region, and vente. The main editor window shows a SQL query (Q5) that calculates the average price per square meter for houses in the Ile-de-France region (R11) with a local code type of 1. The query is as follows:

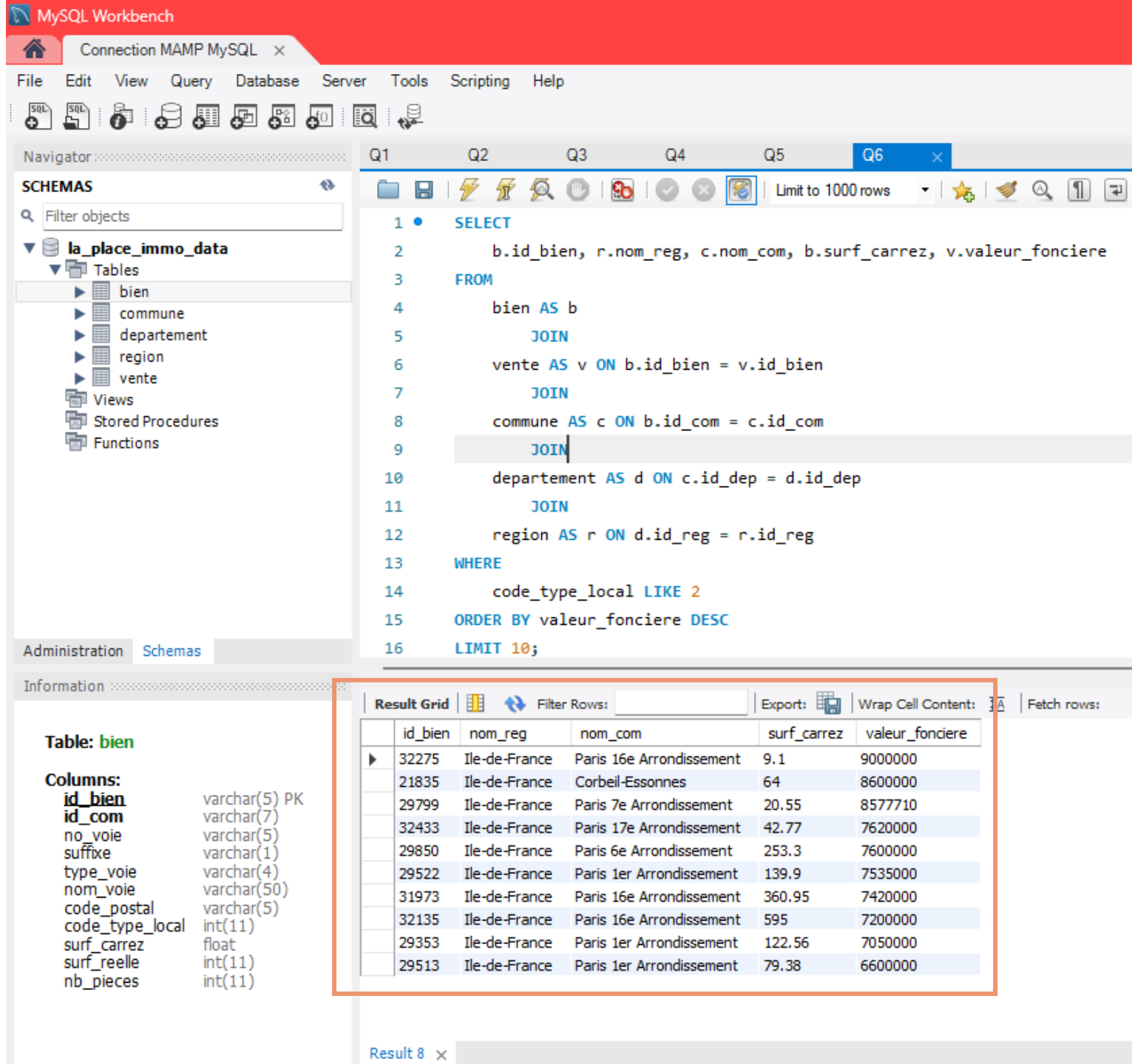
```
SELECT
  r.nom_reg,
  r.id_reg,
  ROUND(AVG(v.valeur_fonciere / b.surf_carrez), 2) AS prix_mcarrez
FROM
  region AS r
  JOIN
    departement AS d ON r.id_reg = d.id_reg
  JOIN
    commune AS c ON d.id_dep = c.id_dep
  JOIN
    bien AS b ON c.id_com = b.id_com
  JOIN
    vente AS v ON b.id_bien = v.id_bien
WHERE
  d.id_reg LIKE 'R11'
  AND code_type_local LIKE 1
GROUP BY r.id_reg;
```

The 'Result Grid' at the bottom shows the output of the query, which is a single row of data:

nom_reg	id_reg	prix_mcarrez
Ile-de-France	R11	3745.01

LISTE DES 10 APPARTEMENTS LES PLUS CHERS
AVEC LA RÉGION ET LE NOMBRE DE MÈTRES CARRÉS.

```
SELECT b.id_bien, r.nom_reg, c.nom_com,
b.surf_carrez, v.valeur_fonciere
FROM bien AS b
JOIN vente AS v ON b.id_bien = v.id_bien
JOIN commune AS c ON b.id_com =
c.id_com JOIN departement AS d ON
c.id_dep = d.id_dep
JOIN region AS r ON d.id_reg = r.id_reg
WHERE code_type_local LIKE 2
ORDER BY valeur_fonciere DESC LIMIT 10;
```



The screenshot shows the MySQL Workbench interface. The top menu bar includes File, Edit, View, Query, Database, Server, Tools, Scripting, and Help. The left sidebar shows the 'SCHEMAS' panel with a tree view of the database 'la_place_immo_data' containing tables: bien, commune, departement, region, and vente. The main editor displays a SQL query (Q6) that selects the top 10 most expensive properties based on their value. The 'Result Grid' at the bottom right shows the results of the query, which are 10 rows of property data.

SQL Query (Q6):

```
SELECT
  b.id_bien, r.nom_reg, c.nom_com, b.surf_carrez, v.valeur_fonciere
FROM
  bien AS b
JOIN
  vente AS v ON b.id_bien = v.id_bien
JOIN
  commune AS c ON b.id_com = c.id_com
JOIN
  departement AS d ON c.id_dep = d.id_dep
JOIN
  region AS r ON d.id_reg = r.id_reg
WHERE
  code_type_local LIKE 2
ORDER BY valeur_fonciere DESC
LIMIT 10;
```

Table: bien

Columns:

- id_bien: varchar(5) PK
- id_com: varchar(7)
- no_voie: varchar(5)
- suffixe: varchar(1)
- type_voie: varchar(4)
- nom_voie: varchar(50)
- code_postal: varchar(5)
- code_type_local: int(11)
- surf_carrez: float
- surf_reelle: int(11)
- nb_pieces: int(11)

Result Grid:

	id_bien	nom_reg	nom_com	surf_carrez	valeur_fonciere
▶	32275	Ile-de-France	Paris 16e Arrondissement	9.1	9000000
	21835	Ile-de-France	Corbeil-Essonnes	64	8600000
	29799	Ile-de-France	Paris 7e Arrondissement	20.55	8577710
	32433	Ile-de-France	Paris 17e Arrondissement	42.77	7620000
	29850	Ile-de-France	Paris 6e Arrondissement	253.3	7600000
	29522	Ile-de-France	Paris 1er Arrondissement	139.9	7535000
	31973	Ile-de-France	Paris 16e Arrondissement	360.95	7420000
	32135	Ile-de-France	Paris 16e Arrondissement	595	7200000
	29353	Ile-de-France	Paris 1er Arrondissement	122.56	7050000
	29513	Ile-de-France	Paris 1er Arrondissement	79.38	6600000

TAUX D'ÉVOLUTION DU NOMBRE DE VENTES ENTRE LE PREMIER ET LE SECOND TRIMESTRE DE 2020.

```
WITH vente_s1 AS (SELECT
ROUND(COUNT(id_vente), 2) AS vente_trimestre_1
FROM vente AS v WHERE v.date_mutation
BETWEEN '2020-01-01' AND '2020-03-31'),
vente_s2 AS (SELECT ROUND(COUNT(id_vente), 2)
AS vente_trimestre_2 FROM vente AS v
WHERE v.date_mutation BETWEEN '2020-04-01'
AND '2020-06-30')
SELECT ROUND(((vente_trimestre_2 -
vente_trimestre_1) / vente_trimestre_1 * 100), 2)
AS evol_ventes FROM vente_s1, vente_s2;
```

DB Browser for SQLite - C:\Users\Clair\Downloads\la_place_immo_db.db

chier Édition Vue Outils Aide

Nouvelle Base de Données Ouvrir une Base de Données Enregistrer les modifications Annuler les modifications Ouvrir u

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

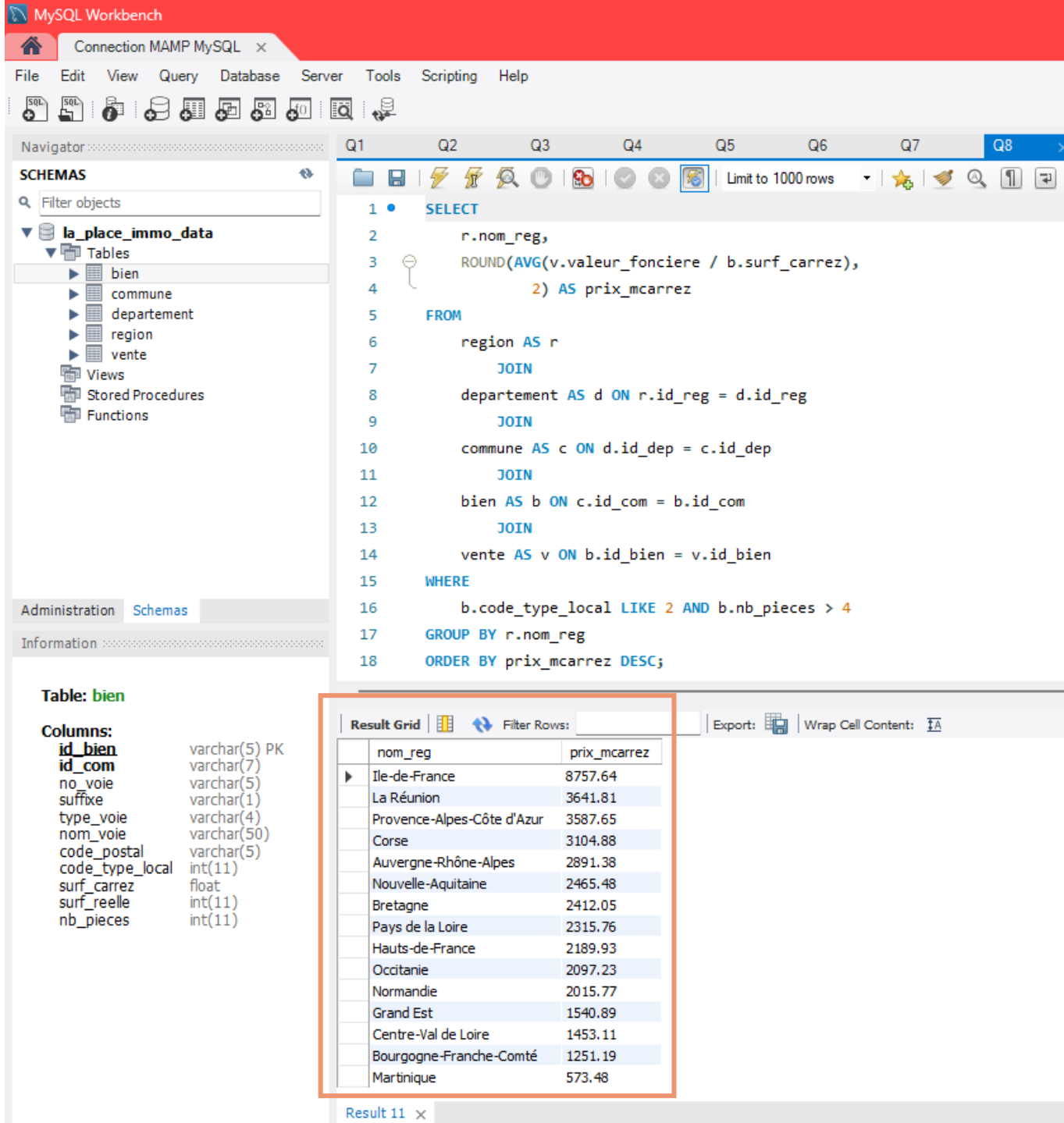
Q7-WITH.sql

```
1 WITH vente_s1 AS (
2     SELECT ROUND(COUNT(id_vente), 2) AS vente_trimestre_1
3     FROM vente AS v
4     WHERE v.date_mutation BETWEEN '2020-01-01' AND '2020-03-31'),
5     vente_s2 AS (
6     SELECT ROUND(COUNT(id_vente), 2) AS vente_trimestre_2
7     FROM vente AS v
8     WHERE v.date_mutation BETWEEN '2020-04-01' AND '2020-06-30')
9     SELECT ROUND(((vente_trimestre_2 - vente_trimestre_1) / vente_trimestre_1 * 100), 2) AS evol_ventes
10    FROM vente_s1, vente_s2;
```

evol_ventes	
1	3.68

CLASSEMENT DES RÉGIONS PAR RAPPORT AU PRIX AU MÈTRE CARRÉ DES APPARTEMENT DE PLUS DE 4 PIÈCES.

```
SELECT r.nom_reg, ROUND(AVG(v.valeur_fonciere
/ b.surf_carrez), 2) AS prix_mcarrez FROM region
AS r JOIN departement AS d ON r.id_reg =
d.id_reg JOIN commune AS c ON d.id_dep =
c.id_dep JOIN bien AS b ON c.id_com = b.id_com
JOIN vente AS v ON b.id_bien = v.id_bien
WHERE b.code_type_local LIKE 2 AND
b.nb_pieces > 4
GROUP BY r.nom_reg
ORDER BY prix_mcarrez DESC;
```



The screenshot shows the MySQL Workbench interface. The top menu bar includes File, Edit, View, Query, Database, Server, Tools, Scripting, and Help. The left sidebar shows the Schemas tree with the database 'la_place_immo_data' expanded, showing tables: bien, commune, departement, region, and vente. The main query editor displays the following SQL query:

```
SELECT
  r.nom_reg,
  ROUND(AVG(v.valeur_fonciere / b.surf_carrez),
    2) AS prix_mcarrez
FROM
  region AS r
  JOIN
    departement AS d ON r.id_reg = d.id_reg
  JOIN
    commune AS c ON d.id_dep = c.id_dep
  JOIN
    bien AS b ON c.id_com = b.id_com
  JOIN
    vente AS v ON b.id_bien = v.id_bien
WHERE
  b.code_type_local LIKE 2 AND b.nb_pieces > 4
GROUP BY r.nom_reg
ORDER BY prix_mcarrez DESC;
```

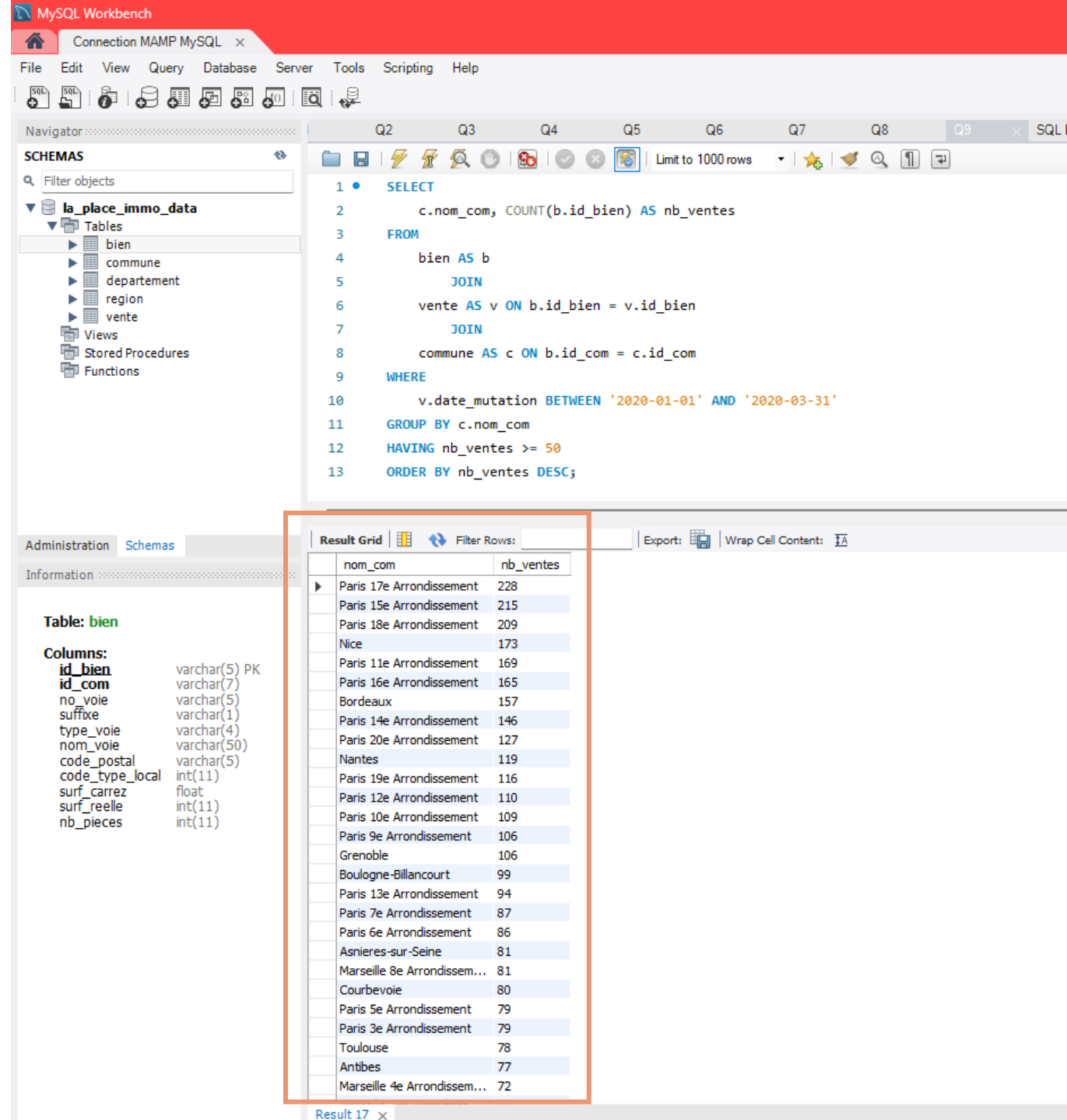
The bottom panel shows the 'Table: bien' structure with columns: id_bien (varchar(5) PK), id_com (varchar(7)), no_voie (varchar(5)), suffixe (varchar(1)), type_voie (varchar(4)), nom_voie (varchar(50)), code_postal (varchar(5)), code_type_local (int(11)), surf_carrez (float), surf_reelle (int(11)), and nb_pieces (int(11)).

The 'Result Grid' shows the following data:

nom_reg	prix_mcarrez
Ile-de-France	8757.64
La Réunion	3641.81
Provence-Alpes-Côte d'Azur	3587.65
Corse	3104.88
Auvergne-Rhône-Alpes	2891.38
Nouvelle-Aquitaine	2465.48
Bretagne	2412.05
Pays de la Loire	2315.76
Hauts-de-France	2189.93
Occitanie	2097.23
Normandie	2015.77
Grand Est	1540.89
Centre-Val de Loire	1453.11
Bourgogne-Franche-Comté	1251.19
Martinique	573.48

LISTE DES COMMUNES
AYANT EU AU MOINS 50 VENTES AU 1ER TRIMESTRE.

```
SELECT c.nom_com,
COUNT(b.id_bien) AS nb_ventes
FROM bien AS b
JOIN vente AS v ON b.id_bien = v.id_bien
JOIN commune AS c ON b.id_com = c.id_com
WHERE v.date_mutation BETWEEN '2020-01-01'
AND '2020-03-31'
GROUP BY c.nom_com
HAVING nb_ventes >= 50
ORDER BY nb_ventes DESC;
```



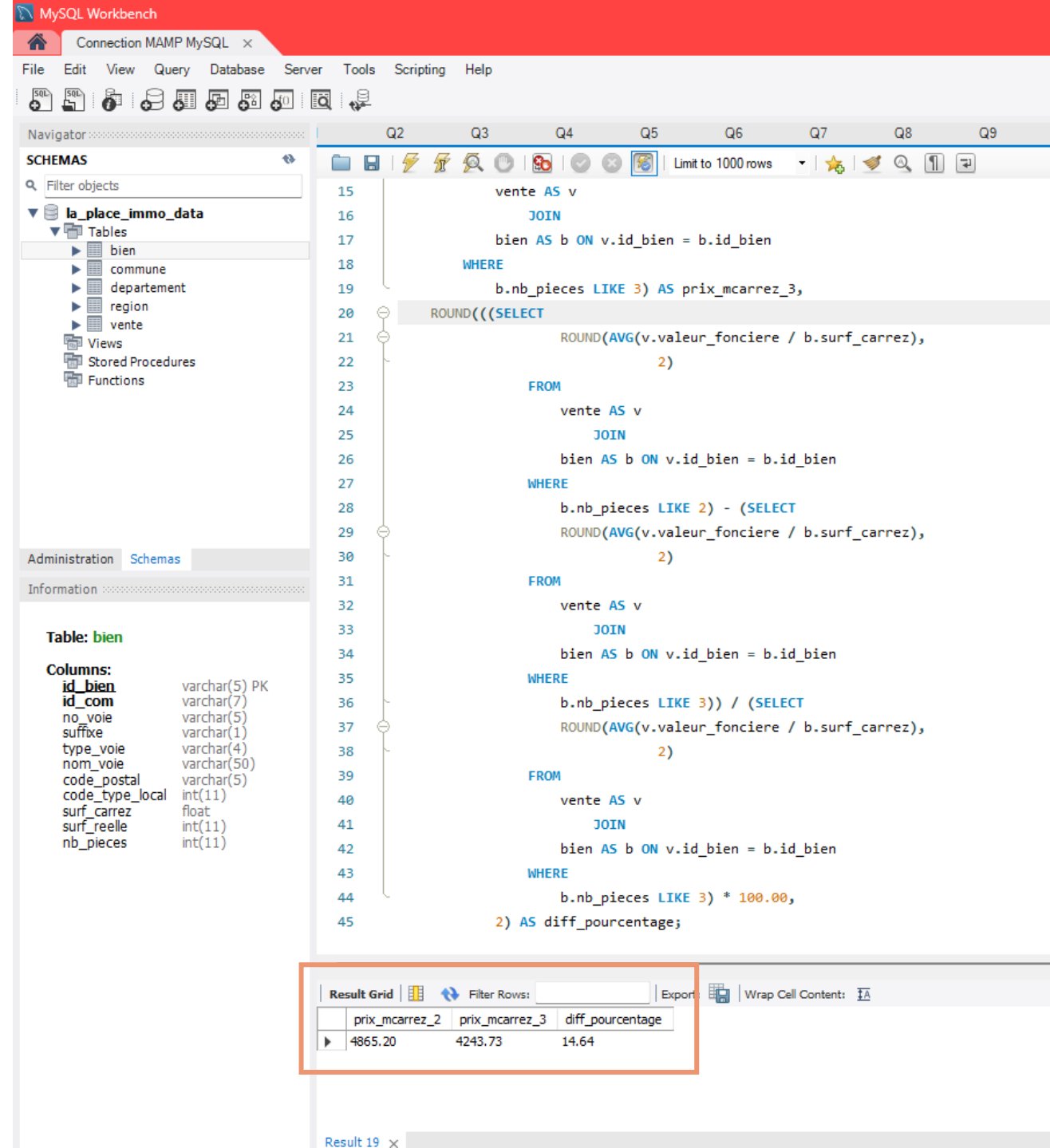
The screenshot shows the MySQL Workbench interface. The top menu bar includes File, Edit, View, Query, Database, Server, Tools, Scripting, and Help. The Navigator pane on the left shows the database structure for 'la_place_immo_data', including tables (bien, commune, departement, region, vente), views, stored procedures, and functions. The main editor displays a SQL query that counts the number of sales (nb_ventes) for each commune (nom_com) where the sale date is between January 1, 2020, and March 31, 2020, and the number of sales is greater than or equal to 50. The results are ordered by the number of sales in descending order. The bottom pane shows the 'Result Grid' with the following data:

nom_com	nb_ventes
Paris 17e Arrondissement	228
Paris 15e Arrondissement	215
Paris 18e Arrondissement	209
Nice	173
Paris 11e Arrondissement	169
Paris 16e Arrondissement	165
Bordeaux	157
Paris 14e Arrondissement	146
Paris 20e Arrondissement	127
Nantes	119
Paris 19e Arrondissement	116
Paris 12e Arrondissement	110
Paris 10e Arrondissement	109
Paris 9e Arrondissement	106
Grenoble	106
Boulogne-Billancourt	99
Paris 13e Arrondissement	94
Paris 7e Arrondissement	87
Paris 6e Arrondissement	86
Asnieres-sur-Seine	81
Marseille 8e Arrondissement	81
Courbevoie	80
Paris 5e Arrondissement	79
Paris 3e Arrondissement	79
Toulouse	78
Antibes	77
Marseille 4e Arrondissement	72

DIFFÉRENCE EN POURCENTAGE DU PRIX AU MÈTRE CARRÉ
ENTRE UN APPARTEMENT DE 2 PIÈCES ET UN APPARTEMENT DE 3 PIÈCES.

```
SELECT ROUND(((SELECT
ROUND(AVG(v.valeur_fonciere / b.surf_carrez),
2) FROM vente AS v JOIN bien AS b ON v.id_bien =
b.id_bien WHERE b.nb_pieces LIKE 2) - (SELECT
ROUND(AVG(v.valeur_fonciere / b.surf_carrez),
2) FROM vente AS v JOIN bien AS b ON v.id_bien =
b.id_bien WHERE b.nb_pieces LIKE 3)) / (SELECT
ROUND(AVG(v.valeur_fonciere / b.surf_carrez),
2) FROM vente AS v JOIN bien AS b ON v.id_bien =
b.id_bien WHERE b.nb_pieces LIKE 3) * 100.00, 2) AS
diff_pourcentage;
```

10/12



The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'Schemas' tree with 'la_place_immo_data' expanded, showing tables like 'bien', 'commune', 'departement', 'region', 'vente', 'Views', 'Stored Procedures', and 'Functions'. The 'Table: bien' structure is shown with columns: id_bien (varchar(5) PK), id_com (varchar(7)), no_voie (varchar(5)), suffixe (varchar(1)), type_voie (varchar(4)), nom_voie (varchar(50)), code_postal (varchar(5)), code_type_local (int(11)), surf_carrez (float), surf_reelle (int(11)), and nb_pieces (int(11)).

The main editor shows a SQL query (lines 15-45) that calculates the percentage difference in price per square meter between 2-piece and 3-piece apartments. The query uses subqueries and the ROUND function to format the results.

The 'Result Grid' at the bottom shows the output of the query:

	prix_mcarrez_2	prix_mcarrez_3	diff_pourcentage
▶	4865.20	4243.73	14.64

MOYENNES DE VALEURS FONCIÈRES POUR LE TOP 3 DES COMMUNES DES DÉPARTEMENTS 6, 13, 33, 59 ET 69.

```
WITH valeur_par_ville AS (
SELECT d.code_dep, c.nom_com, AVG(v.valeur_fonciere) AS valeur
FROM vente AS v JOIN bien AS b ON v.id_bien = b.id_bien
JOIN commune AS c ON b.id_com = c.id_com
JOIN departement AS d ON c.id_dep = d.id_dep
WHERE d.code_dep IN (06,13,33,59,69)
GROUP BY d.code_dep, c.nom_com)
SELECT code_dep AS "Département", nom_com AS "Commune",
ROUND(valeur, 2) AS "Prix moyen" FROM (SELECT code_dep, nom_com,
valeur, rank() OVER (PARTITION BY code_dep ORDER BY valeur DESC)
AS rang FROM valeur_par_ville) AS resultat WHERE rang <= 3;
```

DB Browser for SQLite - C:\Users\Clair\Downloads\OpenClassRooms\Projet 3 - Créez et utilisez une base de données

Fichier Édition Vue Outils Aide

Nouvelle Base de Données Ouvrir une Base de Données Enregistrer les modifications Annuler les modifications

Structure de la Base de Données Parcourir les données Éditer les Pragma Exécuter le SQL

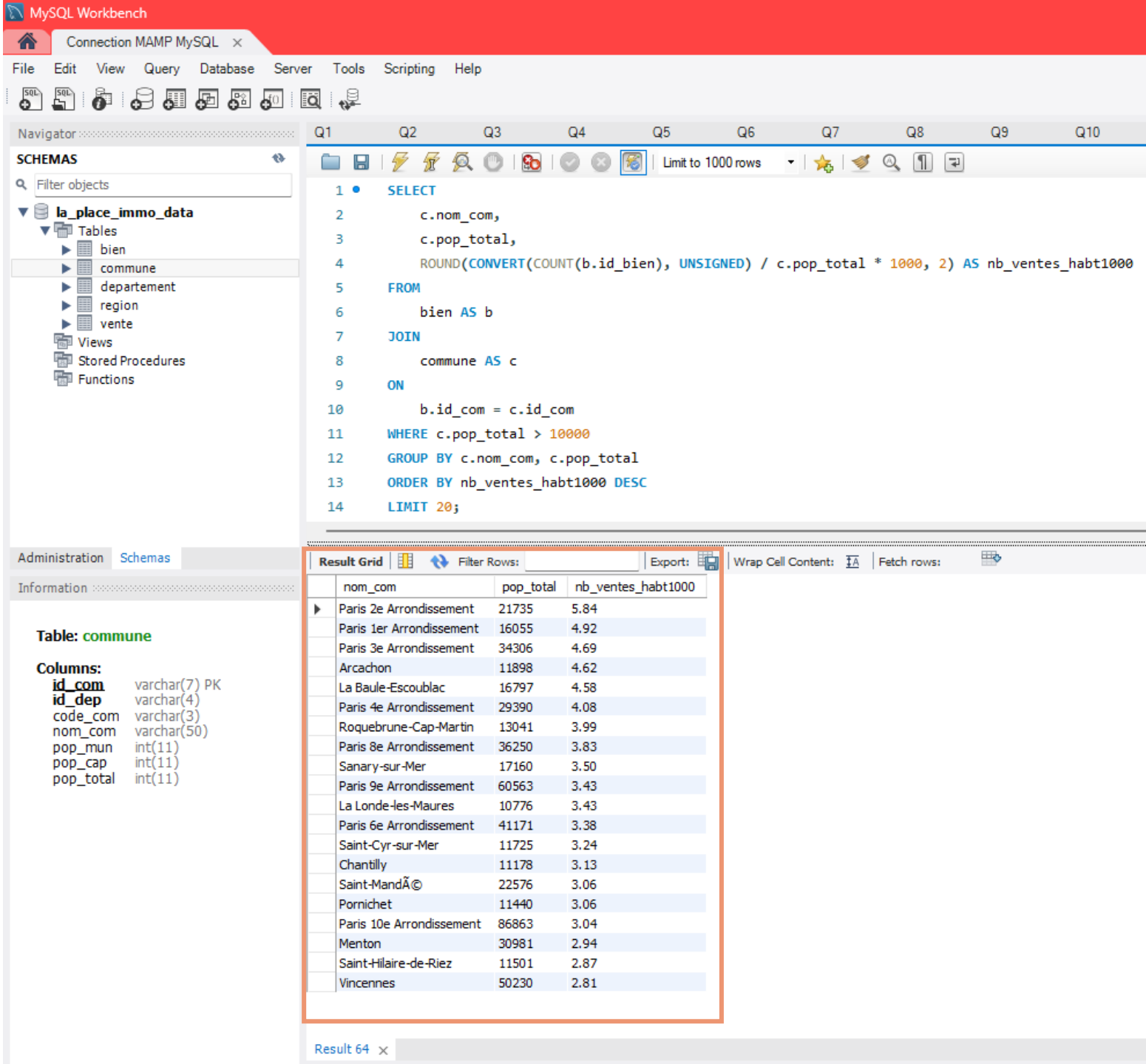
Q7-WITH.sql Q11-ok.sql SQL 3

```
1 WITH valeur_par_ville AS (
2   SELECT d.code_dep, c.nom_com, AVG(v.valeur_fonciere) as valeur
3   FROM vente AS v
4   JOIN bien AS b ON v.id_bien = b.id_bien
5   JOIN commune AS c ON b.id_com = c.id_com
6   JOIN departement AS d ON c.id_dep = d.id_dep
7   WHERE d.code_dep IN ('06', 13, 33, 59, 69)
8   GROUP BY d.code_dep, c.nom_com)
9   SELECT code_dep AS "Département", nom_com AS "Commune",
10  ROUND(valeur, 2) AS "Prix moyen"
11 FROM (
12   SELECT code_dep, nom_com, valeur,
13   rank() OVER (PARTITION BY code_dep ORDER BY valeur DESC) AS rang
14   FROM valeur_par_ville) AS resultat WHERE rang <= 3;
```

	Département	Commune	Prix moyen
1	06	Saint-Jean-Cap-Ferrat	968750.0
2	06	Eze	655000.0
3	06	Mouans-Sartoux	476898.1
4	13	Gignac-la-Nerthe	330000.0
5	13	Saint-Savournin	314425.0
6	13	Cassis	313416.88
7	33	Lège-Cap-Ferret	549500.64
8	33	Vayres	335000.0
9	33	Arcachon	307435.93
10	59	Bersée	433202.0
11	59	Cysoing	408550.0
12	59	Halluin	322250.0

20 COMMUNES AVEC LE PLUS DE TRANSACTIONS
POUR 1000 HABITANTS POUR LES COMMUNES
QUI DÉPASSENT LES 10 000 HABITANTS.

```
SELECT c.nom_com, c.pop_total,
ROUND(CONVERT(COUNT(b.id_bien),
UNSIGNED) / c.pop_total * 1000, 2) AS
nb_ventes_habt1000 FROM bien AS b
JOIN commune AS c ON b.id_com =
c.id_com WHERE c.pop_total > 10000
GROUP BY c.nom_com, c.pop_total
ORDER BY nb_ventes_habt1000 DESC
LIMIT 20;
```



The screenshot shows the MySQL Workbench interface. The top menu bar includes File, Edit, View, Query, Database, Server, Tools, Scripting, and Help. The left sidebar shows the Navigator with the Schemas section expanded, displaying the database structure for 'la_place_immo_data'. The main editor shows a SQL query that calculates the number of transactions per 1000 inhabitants for communes with a population greater than 10,000, ordered by the result in descending order, limited to 20 rows.

The query is as follows:

```
1 SELECT
2   c.nom_com,
3   c.pop_total,
4   ROUND(CONVERT(COUNT(b.id_bien), UNSIGNED) / c.pop_total * 1000, 2) AS nb_ventes_habt1000
5 FROM
6   bien AS b
7 JOIN
8   commune AS c
9 ON
10  b.id_com = c.id_com
11 WHERE c.pop_total > 10000
12 GROUP BY c.nom_com, c.pop_total
13 ORDER BY nb_ventes_habt1000 DESC
14 LIMIT 20;
```

The bottom panel shows the Result Grid with the following data:

nom_com	pop_total	nb_ventes_habt1000
Paris 2e Arrondissement	21735	5.84
Paris 1er Arrondissement	16055	4.92
Paris 3e Arrondissement	34306	4.69
Arcachon	11898	4.62
La Baule-Escoublac	16797	4.58
Paris 4e Arrondissement	29390	4.08
Roquebrune-Cap-Martin	13041	3.99
Paris 8e Arrondissement	36250	3.83
Sanary-sur-Mer	17160	3.50
Paris 9e Arrondissement	60563	3.43
La Londe-les-Maures	10776	3.43
Paris 6e Arrondissement	41171	3.38
Saint-Cyr-sur-Mer	11725	3.24
Chantilly	11178	3.13
Saint-MandÃ©	22576	3.06
Pornichet	11440	3.06
Paris 10e Arrondissement	86863	3.04
Menton	30981	2.94
Saint-Hilaire-de-Riez	11501	2.87
Vincennes	50230	2.81