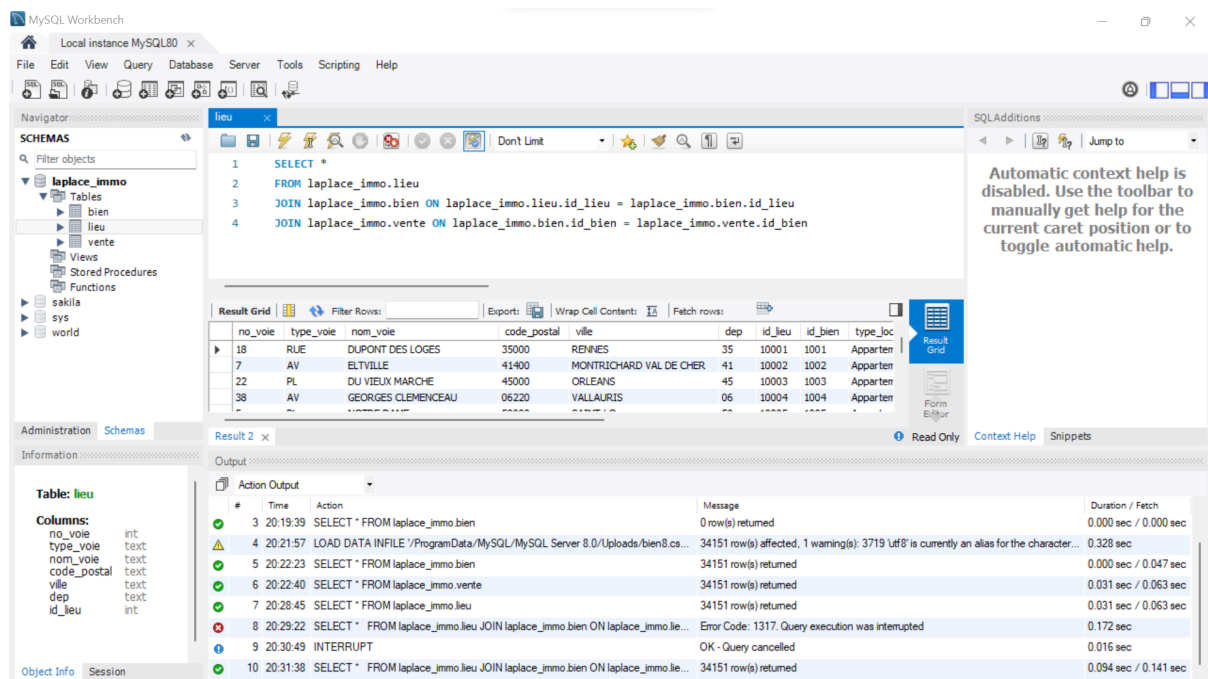


Data Analysis using Mysql

Objectifs du projet:

- 1) Réaliser une jointure de 3 tables (Lieu, Bien et Vente)
- 2) Trouver le nombre d'appartements vendus au 1er semestre 2020
- 3) La proportion des ventes d'appartements par le nombre de pièces.
- 4) Liste des 10 départements où le prix du mètre carré est le plus élevé.
- 5) Prix moyen du mètre carré d'une maison en Île-de-France.
- 6) Liste des 10 appartements les plus chers avec le département et le nombre de mètres carrés.
- 7) Taux d'évolution du nombre de ventes entre le premier et le second trimestre de 2020.
- 8) Liste des communes où le nombre de ventes a augmenté d'au moins 20% entre le premier et le second trimestre de 2020
- 9) Les moyennes de valeurs foncières pour le top 3 des communes des départements 6, 13, 33, 59 et 69

1. Réaliser une jointure de 3 tables (Lieu, Bien et Vente)



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

```
1 SELECT *
2 FROM laplace_immo.lieu
3 JOIN laplace_immo.bien ON laplace_immo.lieu.id_lieu = laplace_immo.bien.id_lieu
4 JOIN laplace_immo.vente ON laplace_immo.bien.id_bien = laplace_immo.vente.id_bien
```

The Results window displays the following data:

	no_voie	type_voie	nom_voie	code_postal	ville	dep	id_lieu	id_bien	type_loc
18	RUE		DUPONT DES LOGES	35000	RENNES	35	10001	1001	Apparten
7	AV		ELTVILLE	41400	MONTRICHARD VAL DE CHER	41	10002	1002	Apparten
22	PL		DU VIEUX MARCHE	45000	ORLEANS	45	10003	1003	Apparten
38	AV		GEORGES CLEMENCEAU	06220	VALLAURIS	06	10004	1004	Apparten

The Output window shows the execution of the query, with the following messages:

```
3 20:19:39 SELECT * FROM laplace_immo bien 0 row(s) returned 0.000 sec / 0.000 sec
4 20:21:57 LOAD DATA INFILE 'C:\ProgramData\MySQL\MySQL Server 8.0\Uploads\bien8.cs...' 34151 row(s) affected, 1 warning(s): 3719 Left's is currently an alias for the character... 0.328 sec
5 20:22:23 SELECT * FROM laplace_immo bien 34151 row(s) returned 0.000 sec / 0.047 sec
6 20:22:40 SELECT * FROM laplace_immo.vente 34151 row(s) returned 0.031 sec / 0.063 sec
7 20:28:45 SELECT * FROM laplace_immo.lieu 34151 row(s) returned 0.031 sec / 0.063 sec
8 20:29:22 SELECT * FROM laplace_immo.lieu JOIN laplace_immo.bien ON laplace_immo.lie... Error Code: 1317. Query execution was interrupted 0.172 sec
9 20:30:49 INTERRUPT OK - Query cancelled 0.016 sec
10 20:31:38 SELECT * FROM laplace_immo.lieu JOIN laplace_immo.bien ON laplace_immo.lie... 34151 row(s) returned 0.094 sec / 0.141 sec
```

2. Nombre total d'appartements vendus au 1er semestre 2020.

```
-- Nombre total d'appartements vendus au 1er semestre 2020.
SELECT
  YEAR(date) AS Année,
  type_local AS Bien,
  IF(MONTH(date) < 7, 1, 2) AS Semestre,
  COUNT(*) as No_ventes
FROM
  laplace_immo.merged
WHERE type_local = 'Appartement' AND YEAR(date)= 2020 AND MONTH(date) < 7
```

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

▼ laplace_immo

Tables

► bien

► lieu

► merged

► vente

Views

Stored Procedures

Functions

► sakila

► sys

► world

Administration Schemas

Information

No object selected

merged x

```

1 -- Nombre total d'appartements vendus au 1er semestre 2020.
2 • SELECT
3     YEAR(date) AS Année,
4     type_local AS Bien,
5     IF(MONTH(date) < 7, 1, 2) AS Semestre,
6     COUNT(*) as No_ventes
7 FROM
8     laplace_immo.merged
9 WHERE type_local = 'Appartement' AND YEAR(date)= 2020 AND MONTH(date) < 7

```

Result Grid

Filter Rows:

Export: Wrap Cell Content: 15

Année	Bien	Semestre	No_ventes
2020	Appartement	1	31362

Result 10 x

Output

Read Only

3. Proportion des ventes d'appartements par le nombre de pièces.

```

--Proportion des ventes d'appartements par le nombre de pièces.
SELECT type_local AS Bien,
       no_pieces,
       ROUND(COUNT(*) *100 / (SELECT COUNT(*) FROM laplace_immo.merged),1) AS '% Ventres'
FROM laplace_immo.merged
WHERE type_local = 'Appartement'
GROUP BY no_pieces
ORDER BY no_pieces

```

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

▼ laplace_immo

Tables

► bien

► lieu

► merged

► vente

Views

Stored Procedures

Functions

► sakila

► sys

► world

Administration Schemas

Information

No object selected

merged x

```

1 -- Proportion des ventes d'appartements par le nombre de pièces.
2 • SELECT type_local AS Bien,
3     no_pieces,
4     ROUND(COUNT(*) *100 / (SELECT COUNT(*) FROM laplace_immo.merged),1) AS '% Ventres'
5 FROM laplace_immo.merged
6 WHERE type_local = 'Appartement'
7 GROUP BY no_pieces
8 ORDER BY no_pieces

```

Result Grid

Filter Rows:

Export: Wrap Cell Content: 15

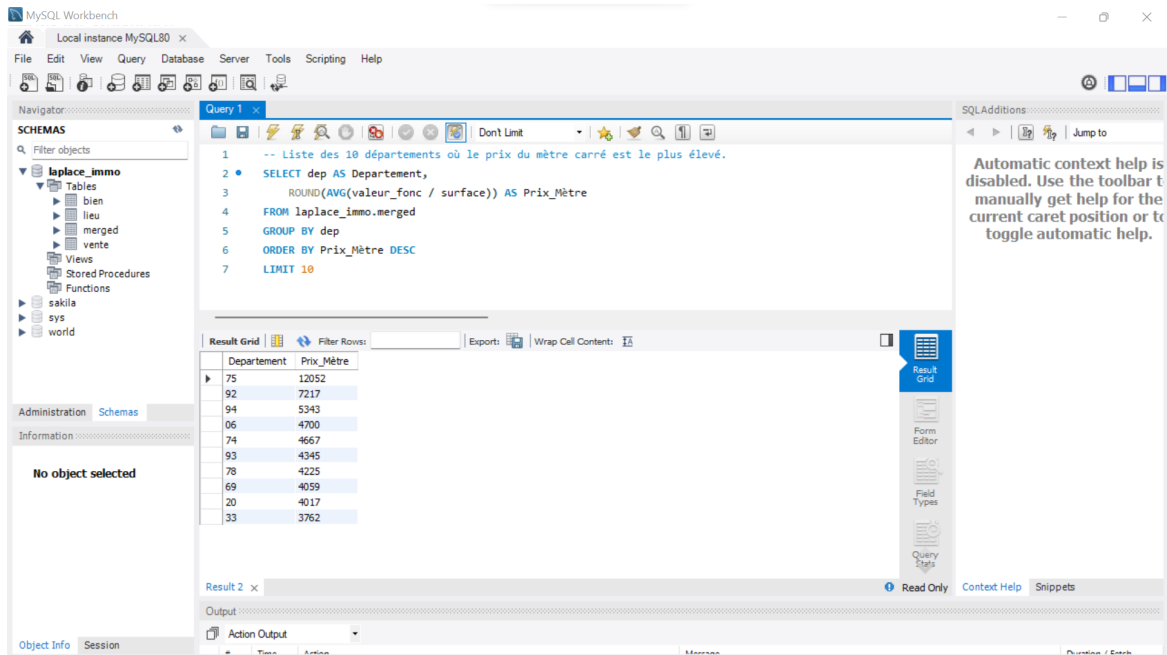
Bien	no_pieces	% Ventres
Appartement	0	0.1
Appartement	1	19.7
Appartement	2	28.6
Appartement	3	26.3
Appartement	4	13.1
Appartement	5	3.3
Appartement	6	0.6
Appartement	7	0.2
Appartement	8	0.0
Appartement	9	0.0
Appartement	10	0.0
Appartement	11	0.0

Result 6 x

Read Only

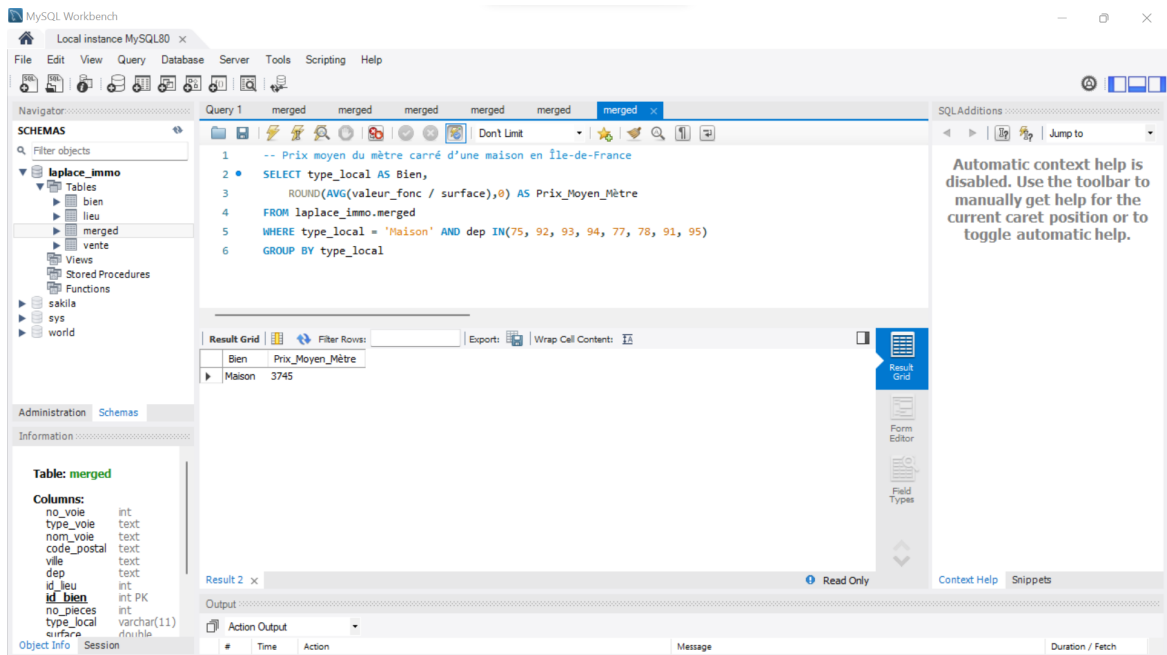
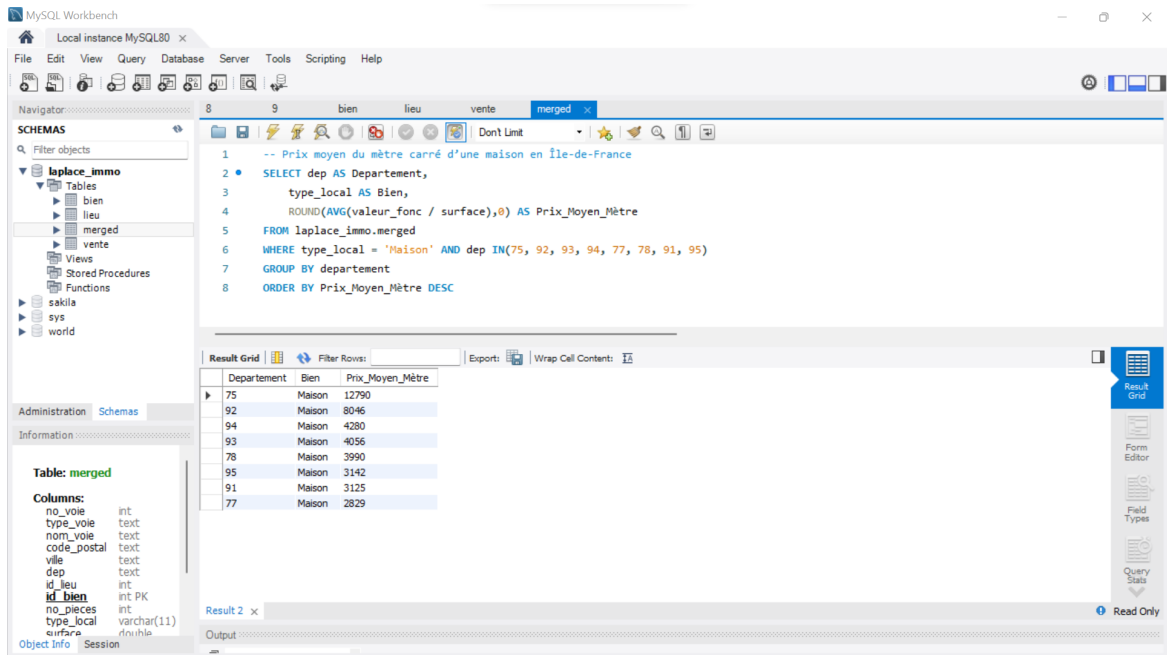
4. Liste des 10 départements où le prix du mètre carré est le plus élevé.

```
-- Liste des 10 départements où le prix du mètre carré est le plus élevé.
SELECT dep AS Departement,
       ROUND(AVG(valeur_fonc / surface)) AS Prix_Mètre
FROM laplace_immo.merged
GROUP BY dep
ORDER BY Prix_Mètre DESC
LIMIT 10
```



5. Prix moyen du mètre carré d'une maison en Île-de-France.

```
-- Prix moyen du mètre carré d'une maison en Île-de-France
SELECT dep AS Departement,
       type_local AS Bien,
       ROUND(AVG(valeur_fonc / surface),0) AS Prix_Moyen_Mètre
FROM laplace_immo.merged
WHERE type_local = 'Maison' AND dep IN(75, 92, 93, 94, 77, 78, 91, 95)
GROUP BY departement
ORDER BY Prix_Moyen_Mètre DESC
```

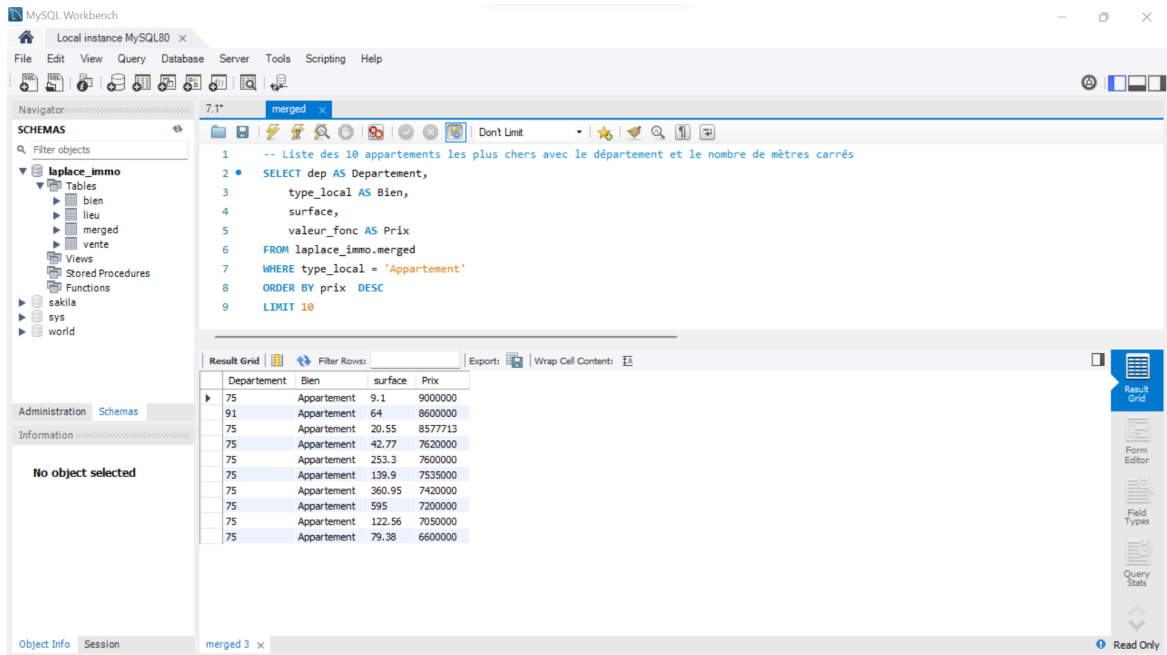


6. Liste des 10 appartements les plus chers avec le département et le nombre de mètres carrés.

```

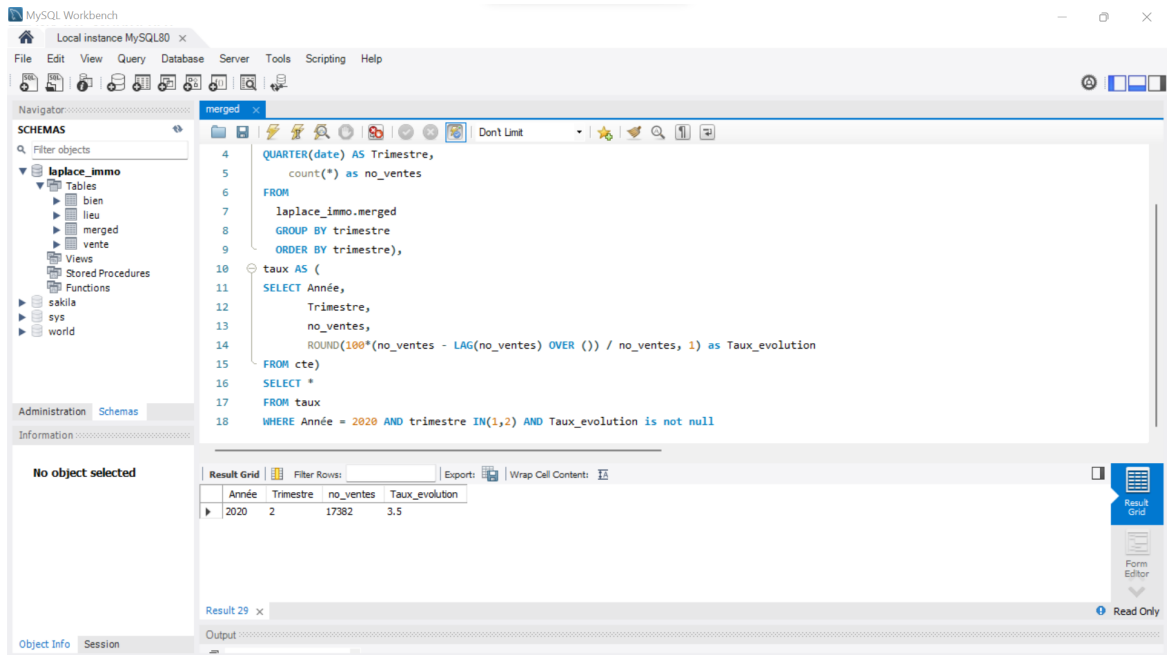
-- Liste des 10 appartements les plus chers avec le département et le nombre de mètres carrés
SELECT dep AS Departement,
       type_local AS Bien,
       surface,
       valeur_fonc AS Prix
FROM laplace_immo.merged
WHERE type_local = 'Appartement'
ORDER BY prix DESC
LIMIT 10

```



7. Taux d'évolution du nombre de ventes entre le premier et le second trimestre de 2020.

```
-- Taux d'évolution du nombre de ventes entre le premier et le second trimestre de 2020.
WITH cte AS (
  SELECT YEAR(date) AS Année,
  QUARTER(date) AS Trimestre,
  count(*) as no_ventes
  FROM
    laplace_immo.merged
  GROUP BY trimestre
  ORDER BY trimestre),
taux AS (
  SELECT Année,
  Trimestre,
  no_ventes,
  ROUND(100*(no_ventes - LAG(no_ventes) OVER ()) / no_ventes, 1) as Taux_evolution
  FROM cte)
SELECT *
FROM taux
WHERE Année = 2020 AND trimestre IN(1,2) AND Taux_evolution is not null
```



8. Liste des communes où le nombre de ventes a augmenté d'au moins 20% entre le premier et le second trimestre de 2020

```

-- Liste des communes où le nombre de ventes a augmenté d'au moins 20% entre le premier et le second trimestre de 2020
WITH
cte AS (
  SELECT ville,
    YEAR(date) AS Année,
    QUARTER(date) AS Trimestre,
    COUNT(*) AS No_ventes
  FROM laplace_immo.merged
  group by ville, trimestre
  order by ville ),
cte_taux AS (
  SELECT *,
  ROUND(100*(no_ventes - LAG(no_ventes) OVER (PARTITION BY ville)) / LAG(no_ventes) OVER (), 1) as taux_evolution
  FROM cte )

SELECT ville, Année, Trimestre, taux_evolution
FROM cte_taux
WHERE taux_evolution > 20 AND Trimestre = 2
ORDER BY ville

```

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' tree with 'laplace_immo' selected. The main editor contains a SQL query that calculates the percentage change in sales volume over time for different cities. The query uses two Common Table Expressions (CTEs): 'cte AS' to calculate sales volume by city and quarter, and 'cte_taux AS' to calculate the percentage change using the LAG function. The final result is ordered by city.

```

1 WITH
2   cte AS (
3     SELECT ville,
4            YEAR(date) AS Année,
5            QUARTER(date) AS Trimestre,
6            COUNT(*) AS No_ventes
7     FROM laplace_immo.merged
8     GROUP BY ville, trimestre
9     ORDER BY ville ),
10  cte_taux AS (
11    SELECT *,
12    ROUND(100*(no_ventes - LAG(no_ventes) OVER (PARTITION BY ville)) / LAG(no_ventes) OVER (), 1) AS taux_evolution
13    FROM cte )
14
15  SELECT ville, Année, Trimestre, taux_evolution
16  FROM cte_taux
17  WHERE taux_evolution > 20 AND Trimestre = 2
18  ORDER BY ville

```

The 'Result Grid' at the bottom shows the following data:

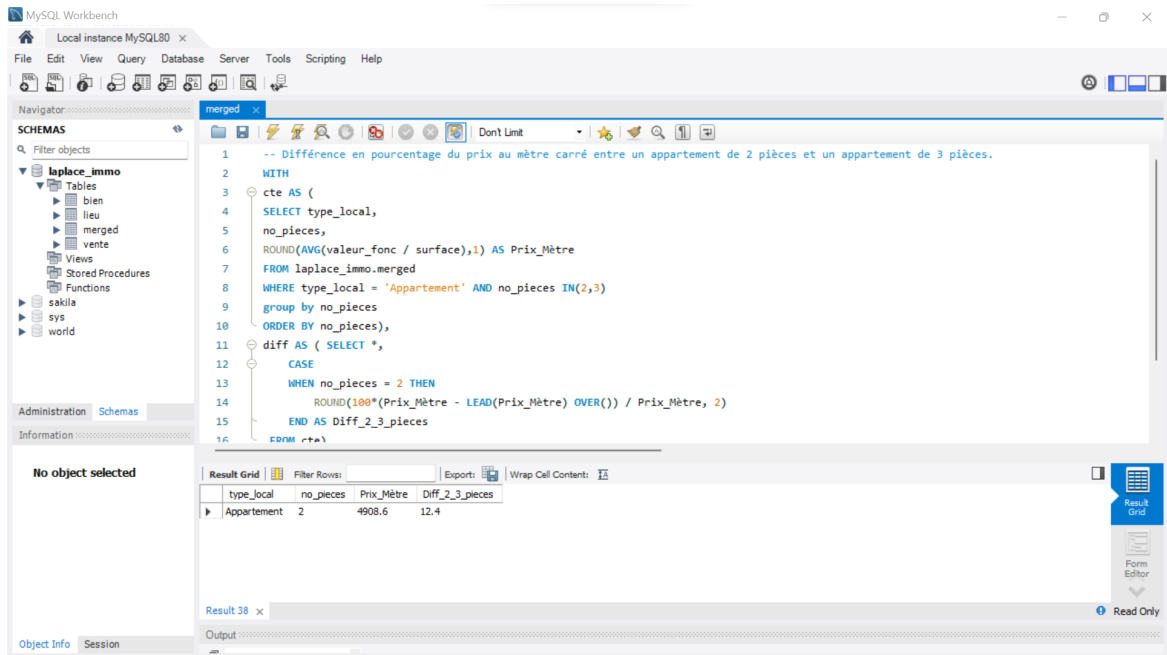
ville	Année	Trimestre	taux_evolution
ABBEVILLE	2020	2	350.0
ABLON-SUR-SEINE	2020	2	160.0
AGDE	2020	2	104.5
AIGUES-MORTES	2020	2	83.3
AIRE-SUR-L'ADOUR	2020	2	400.0
ALENÇON	2020	2	100.0

9. Différence en pourcentage du prix au mètre carré entre un appartement de 2 pièces et un appartement de 3 pièces.

```

-- Différence en pourcentage du prix au mètre carré entre un appartement de 2 pièces et un appartement de 3 pièces.
WITH
cte AS (
SELECT type_local,
no_pieces,
ROUND(AVG(valeur_fonc / surface),1) AS Prix_Mètre
FROM laplace_immo.merged
WHERE type_local = 'Appartement' AND no_pieces IN(2,3)
GROUP BY no_pieces
ORDER BY no_pieces),
diff AS ( SELECT *,
CASE
WHEN no_pieces = 2 THEN
ROUND(100*(Prix_Mètre - LEAD(Prix_Mètre) OVER()) / Prix_Mètre, 2)
END AS Diff_2_3_pieces
FROM cte)
SELECT *
FROM diff
WHERE Diff_2_3_pieces is not null

```



10. Les moyennes de valeurs foncières pour le top 3 des communes des départements 6, 13, 33, 59 et 69

```

-- Les moyennes de valeurs foncières pour le top 3 des communes des départements 6, 13, 33, 59 et 69
WITH
cte AS (
SELECT dep,
ville,
ROUND(AVG(valeur_fonc),1) as Prix_Moyen
FROM laplace_immo.merged
WHERE dep IN('06', '13', '33', '59', '69')
GROUP BY ville, dep
ORDER BY dep, Prix_Moyen DESC )
SELECT dep, ville, Prix_Moyen
FROM
(SELECT *,
@count := IF(@dep = dep, @count + 1, 1) AS count,
@dep := dep
FROM cte
) top_villes
WHERE count <= 3;

```


MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

laplace_immo

Tables

bien

lieu

merged

vente

Views

Stored Procedures

Functions

sakila

sys

world

Administration Schemas

Information

No object selected

```

2 • WITH
3   cte AS (
4     SELECT dep,
5            ville,
6            ROUND(AVG(valeur_fonc),1) as Prix_Moyen
7     FROM laplace_immo.merged
8     WHERE dep IN('06', '13', '33', '59', '69')
9     GROUP BY ville, dep
10    ORDER BY dep, Prix_Moyen DESC )
11    SELECT dep, ville, Prix_Moyen
12  FROM
13    (SELECT *,
14     @count := IF(@dep = dep, @count + 1, 1) AS count,
15     @dep := dep
16    FROM cte ) top_villes
17   WHERE count <= 3;

```

Result Grid

dep	ville	Prix_Moyen
06	SAINT-JEAN-CAP-FERRAT	968750
06	EZE	655000
06	MOUANS-SARTOULX	476898.1
13	GIGNAC-LA-NERTHE	330000
13	SAINT SAVOURNIN	314425
13	CASSIS	313416.9

Result 39 x

Output

Read Only