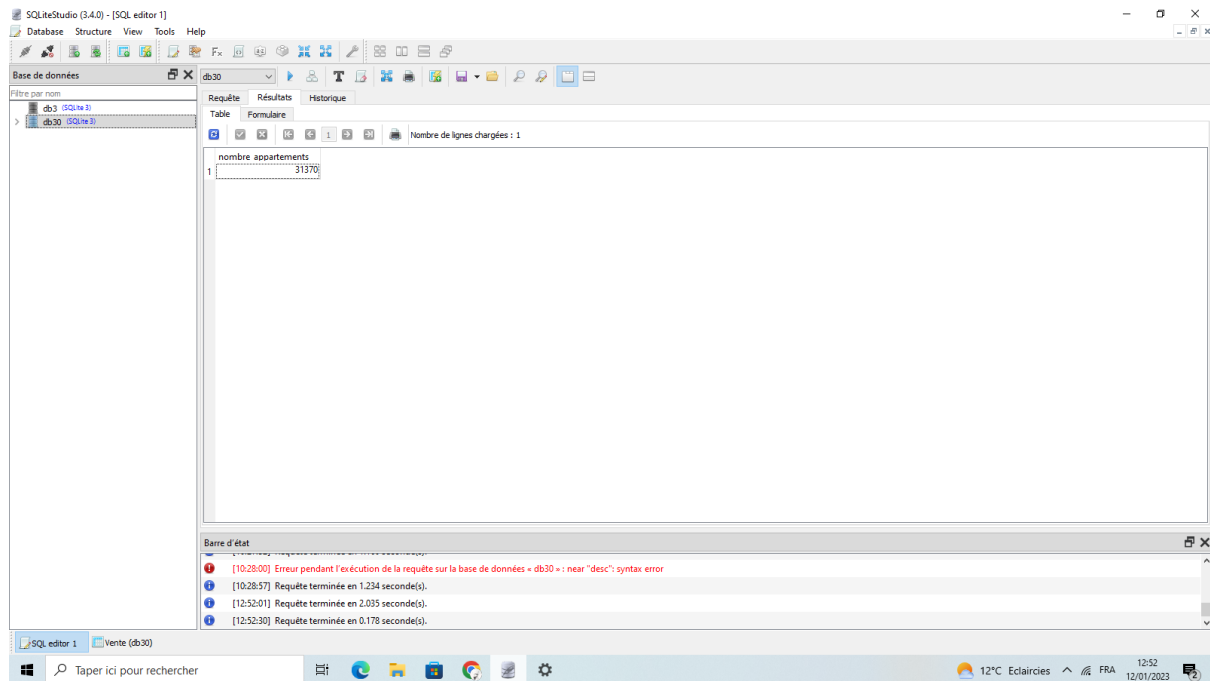


```
--1 Nombre total d'appartements vendus au 1er semestre 2020
select
count(*) as nombre_appartements
from Vente v, Bien b
where b.id_bien = v.id_bien
and v.date>='2020-01-01' and v.date<'2020-07-01'
and b.Type_local="Appartement"
```



--2 Le nombre de ventes d'appartement par région pour le 1er semestre 2020

```
select
r.nom_region,
count(v.id_vente) as nombre_de_ventes
from
Region r, Departement d, Commune c, Bien b, Vente v
where
r.id_code_region = d.id_code_region
and d.id_code_departement = c.id_code_departement
and c.id_code_dep_commune = b.id_code_dep_commune
and b.id_bien = v.id_bien
and v.date>='2020-01-01' and v.date<'2020-07-01'
and b.Type_local="Appartement"
group by
r.nom_region
order by count(v.id_vente) desc
```

SQLStudio (3.4.0) - [SQL editor 1]

Database Structure View Tools Help

Base de données

Requête Résultats Historique

Table Formulaire

Nombre de lignes chargées : 17

	nom region	nombre de ventes
1	ILE-DE-FRANCE	13979
2	PROVENCE-ALPES-COTE D'AZUR	3650
3	AUVERGNE-RHONE-ALPES	3254
4	NOUVELLE-AQUITAINE	1934
5	OCCITANIE	1641
6	PAYS DE LA LOIRE	1358
7	HAUTS-DE-FRANCE	1255
8	GRAND-EST	984
9	BRETAGNE	983
10	NORMANDIE	863
11	CENTRE-VAL DE LOIRE	696
12	BOURGOGNE-FRANCHE-COMTE	376
13	CORSE	223
14	MARTINIQUE	64
15	LA REUNION	44
16	GUYANE	34
17	GUADELOUPE	2

Barre d'état

[10:26:19] Requête terminée en 9.311 seconde(s).

[10:27:32] Requête terminée en 1.160 seconde(s).

[10:28:00] Erreur pendant l'exécution de la requête sur la base de données « db30 » : near "desc": syntax error

[10:28:57] Requête terminée en 1.234 seconde(s).

SQL editor 1 Vente (db30)

Taper ici pour rechercher

12°C Eclaircies FRA 12:49 12/01/2023

--3 Proportion des ventes d'appartement par le nombre des pieces

WITH

table_1 as

(select

count(*) as Total_ventes

from Vente v, Bien b

where b.Type_local="Appartement"

and b.id_bien = v.id_bien

)

select

table_1.Total_ventes,

b.Total_pieces,

count(*) as Ventes_par_pieces,

round(100.0*count(*)/table_1.Total_ventes, 3) as taux_en_pourcentage

from

Vente v, Bien b, table_1

where

b.Type_local="Appartement"

and b.id_bien = v.id_bien

group by

b.Total_pieces

The screenshot shows the SQLStudio 3.4.0 interface. The left pane displays the database structure for 'db3_1' (SQLite 3), including tables like 'Bien', 'Commune', 'Departement', 'Region', 'Vente', and 'Vente_2'. The main editor shows a SQL query that calculates the proportion of apartment sales by the number of pieces. The query is as follows:

```
1 from Vente v, Bien b
2 where b.Type_local="Appartement"
3 and b.id_bien = v.id_bien
4
5
6
7
8
9
10
11 select
12 table_1.Total_ventes,
13 b.Total_pieces,
14 count(*) as Ventes_par_pieces,
15 round(100.0*count(*)/table_1.Total_ventes, 3) as taux_en_pourcentage
16 from
17 Vente v, Bien b, table_1
18 where
19 b.Type_local="Appartement"
20 and b.id_bien = v.id_bien
21 group by
22 b.Total_pieces
23
```

The results are displayed in a table with 12 rows and 4 columns: 'Total ventes', 'Total pieces', 'Ventes par pieces', and 'taux en pourcentage'. The data is as follows:

	Total ventes	Total pieces	Ventes par pieces	taux en pourcentage
1	31370	0	30	0.096
2	31370	1	6738	21.479
3	31370	2	9778	31.17
4	31370	3	8962	28.569
5	31370	4	4461	14.221
6	31370	5	1115	3.554
7	31370	6	204	0.65
8	31370	7	54	0.172
9	31370	8	17	0.054
10	31370	9	8	0.026
11	31370	10	2	0.006
12	31370	11	1	0.003

The status bar at the bottom indicates that the query was terminated in 0.392 seconds.

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

select

d.Nom_departement,

round(avg(v.valeur)/avg(b.surface_car),2) as prix_m2

from Region r, Departement d, Commune c, Bien b, Vente v

where

r.id_code_region = d.id_code_region

and d.id_code_departement = c.id_code_departement

and c.id_code_dep_commune = b.id_code_dep_commune

and b.id_bien = v.id_bien

group by

d.Nom_departement

order by

round(avg(v.valeur)/avg(b.surface_car),2) desc

limit 10;

SQLiteStudio (3.4.0) - [SQL editor 1]

Database Structure View Tools Help

Base de données db30

Requête Historique

```
1 --4 Liste des 10 départements où le prix du mètre carré est le plus élevé
2 select
3 d.Nom_departement,
4 round(avg(v.valeur)/avg(b.surface_car),2) as prix_m2
5 from Region r, Departement d, Commune c, Bien b, Vente v
6 where
7 r.id_code_region = d.id_code_region
8 and d.id_code_departement = c.id_code_departement
9 and c.id_code_dep_commune = b.id_code_dep_commune
10 and b.id_bien = v.id_bien
11 group by
12 d.Nom_departement
13 order by
14 round(avg(v.valeur)/avg(b.surface_car),2) desc
15 limit 10;
16
17
```

Table Formulaire

Nombre de lignes chargées : 10

	Nom departement	prix m2
1	PARIS	11891.09
2	LA REUNION	7492.47
3	HAUTS-DE-SEINE	7237.57
4	GUADELOUPE	5594.66
5	VAL-DE-MARNE	4811.4
6	ALPES-MARITIMES	4584.39
7	HAUTE-SAVOIE	4098.99
8	SEINE-SAINT-DENIS	4096.8
9	YVELINES	3974.72
10	RHONE	3875.8

Barre d'état

- [09:10:15] Could not check for updates (TLS initialization failed).
- [09:15:53] Number of rows per page was decreased to 4166 due to number of columns (12) in the data view.
- [09:16:06] Number of rows per page was decreased to 4166 due to number of columns (12) in the data view.
- [09:19:52] Requête terminée en 14.095 seconde(s).
- [09:32:05] Requête terminée en 13.003 seconde(s).

SQL editor 1 Vente (db30)

Taper ici pour rechercher

8°C Ens

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

select

r.nom_region,

b.Type_local,

round(avg(v.valeur)/avg(b.surface_car),2) as prix_de_m2

from

Region r, Departement d, Commune c, Bien b, Vente v

where

r.id_code_region = d.id_code_region

and d.id_code_departement = c.id_code_departement

and c.id_code_dep_commune = b.id_code_dep_commune

and b.id_bien = v.id_bien

and b.Type_local="Maison"

and r.nom_region="ILE-DE-FRANCE";

The screenshot shows the SQLiteStudio 3.4.0 interface. The left sidebar displays the database structure for 'db30', including tables like 'Bien', 'Commune', 'Departement', 'Region', 'Vente', and 'Vues'. The main window is divided into three panes. The top pane, 'Requête', contains the SQL query:

```
1 --5 Prix moyen du mètre carré d'une maison en Ile-de France
2 select
3 r.nom_region,
4 b.type_local,
5 round(avg(v.valeur)/avg(b.surface_car),2) as prix_de_m2
6 from
7 region r, Departement d, Commune c, Bien b, Vente v
8 where
9 r.id_code_region = d.id_code_region
10 and d.id_code_departement = c.id_code_departement
11 and c.id_code_dep_commune = b.id_code_dep_commune
12 and b.id_bien = v.id_bien
13 and b.type_local="Maison"
14 and r.nom_region="ILE-DE-FRANCE";
15
```

 The middle pane, 'Table', shows the results of the query in a table with three columns: 'nom region', 'Type local', and 'prix de m2'. The first row shows 'ILE-DE-FRANCE', 'Maison', and '2234.99'. The bottom pane, 'Barre d'état', displays status messages:

- [11:45:26] Could not check for updates (TLS initialization failed).
- [11:49:06] Number of rows per page was decreased to 4166 due to number of columns (12) in the data view.
- [11:53:26] La base de données passée en paramètre dans la ligne de commande (C:\Users\annag\Documents\Data analyst\Projet3\db3.db) a été temporairement ajoutée à la liste sous le nom : db30
- [11:58:29] Requête terminée en 4.117 seconde(s).

--6 Liste des 10 appartements les plus chers avec la région et le nombre de mètres carrés

select

r.nom_region,

b.Surface_car,

v.valeur

from

Region r, Departement d, Commune c, Bien b, Vente v

where

r.id_code_region = d.id_code_region

and d.id_code_departement = c.id_code_departement

and c.id_code_dep_commune = b.id_code_dep_commune

and b.id_bien = v.id_bien

and b.Type_local='Appartement'

and v.valeur !=''

order by

v.valeur desc

limit 10;

SQLiteStudio (3.4.0) - [Éditeur SQL 1]

Database Structure View Tools Help

Base de données

Requête Historique

1 --6 Liste des 10 appartements les plus chers avec la région et le nombre de mètres carrés
2 select
3 r.nom_region,
4 b.Surface_car,
5 v.valeur
6 from
7 region r, Departement d, Commune c, Bien b, Vente v
8 where
9 r.id_code_region = d.id_code_region
10 and d.id_code_departement = c.id_code_departement
11 and c.id_code_dep_commune = b.id_code_dep_commune
12 and b.id_bien = v.id_bien
13 and b.Type_local='Appartement'
14 and v.valeur !=''
15 order by
16 v.valeur desc
17 limit 10;
18

Table Formulaire

Nombre de lignes chargées : 10

	nom region	Surface car	valeur
1	ILE-DE-FRANCE	111.38	9000000
2	ILE-DE-FRANCE	54.36	8600000
3	ILE-DE-FRANCE	22.03	8577713
4	ILE-DE-FRANCE	69.94	7620000
5	ILE-DE-FRANCE	11.5	7600000
6	ILE-DE-FRANCE	8.2	7535000
7	ILE-DE-FRANCE	49	7420000
8	ILE-DE-FRANCE	41.18	7200000
9	ILE-DE-FRANCE	55.69	7050000
10	ILE-DE-FRANCE	134.28	6600000

Barre d'état

[20:00:39] Requête terminée en 0.469 seconde(s).
[20:00:51] Requête terminée en 0.474 seconde(s).
[20:02:09] Requête terminée en 0.469 seconde(s).

SQL editor 1 Éditeur SQL 1

Taper ici pour rechercher

11°C Nuageux FRA 20:03 05/01/2023

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

with tab1 as (select

count(v.id_vente) as nombre1,

v.date

from Region r, Departement d, Commune c, Bien b, Vente v

where r.id_code_region = d.id_code_region

and d.id_code_departement = c.id_code_departement

and c.id_code_dep_commune = b.id_code_dep_commune

and b.id_bien = v.id_bien

and (v.date>='2020-01-01' and v.date<'2020-04-01')),

tab2 as (select

count(v.id_vente) as nombre2,

v.date

from Region r, Departement d, Commune c, Bien b, Vente v

where r.id_code_region = d.id_code_region

and d.id_code_departement = c.id_code_departement

and c.id_code_dep_commune = b.id_code_dep_commune

and b.id_bien = v.id_bien

and (v.date>='2020-04-01' and v.date<'2020-07-01'))

select

tab1.nombre1,

tab2.nombre2,

round(tab2.nombre2*100.0/tab1.nombre1-100, 2) as taux_pourcentage

from tab1, tab2

The screenshot shows the SQLStudio 3.4.0 interface. The main window displays a SQL query in the 'Requête' tab. The query is a complex SELECT statement with two subqueries (tab1 and tab2) and a final SELECT clause. The query is executed, and the results are shown in the 'Table' tab. The results table has three columns: 'nombre1', 'nombre2', and 'taux_pourcentage'. The first row shows values 16776, 17393, and 3.68 respectively. The status bar at the bottom indicates that the query was terminated in 0.400 seconds.

```
1 --7 Taux d'évolution du nombre de ventes entre le premier et le second trimestre de 2020
2 with tab1 as (select
3 count(v.id_vente) as nombre1,
4 v.date
5 from Region r, Departement d, Commune c, Bien b, Vente v
6 where r.id_code_region = d.id_code_region
7 and d.id_code_departement = c.id_code_departement
8 and c.id_code_dep_commune = b.id_code_dep_commune
9 and b.id_bien = v.id_bien
10 and (v.date>='2020-01-01' and v.date<'2020-04-01')),
11 tab2 as (select
12 count(v.id_vente) as nombre2,
13 v.date
14 from Region r, Departement d, Commune c, Bien b, Vente v
15 where r.id_code_region = d.id_code_region
16 and d.id_code_departement = c.id_code_departement
17 and c.id_code_dep_commune = b.id_code_dep_commune
18 and b.id_bien = v.id_bien
19 and (v.date>='2020-04-01' and v.date<'2020-07-01'))
20 select
21 tab1.nombre1,
22 tab2.nombre2,
23 round(tab2.nombre2*100.0/tab1.nombre1-100, 2) as taux_pourcentage
24 from tab1, tab2
25
```

nombre1	nombre2	taux_pourcentage
16776	17393	3.68

Barre d'état
[12:17:31] Requête terminée en 0.400 seconde(s).
[12:17:39] Requête terminée en 0.401 seconde(s).
[12:17:56] Requête terminée en 0.400 seconde(s).

--8 Le classement des régions par rapport au prix au mètre carré des appartements de plus de 4 pièces

```
select
r.nom_region,
round(avg(v.valeur)/avg(b.Surface_car), 2) as prix_metre2
from Region r, Departement d, Commune c, Bien b, Vente v
where r.id_code_region = d.id_code_region
and d.id_code_departement = c.id_code_departement
and c.id_code_dep_commune = b.id_code_dep_commune
and b.id_bien = v.id_bien
and b.Type_local='Appartement'
and b.Total_pieces>4
group by r.nom_region
order by avg(v.valeur)/avg(b.Surface_car) desc
```

The screenshot shows the SQLiteStudio 3.4.0 interface. The SQL editor contains the following query:

```
1 --8 Le classement des régions par rapport au prix au mètre carré des appartements de plus de 4 pièces
2 select
3 r.nom_region,
4 round(avg(v.valeur)/avg(b.Surface_car), 2) as prix_metre2
5 from Region r, Departement d, Commune c, Bien b, Vente v
6 where r.id_code_region = d.id_code_region
7 and d.id_code_departement = c.id_code_departement
8 and c.id_code_dep_commune = b.id_code_dep_commune
9 and b.id_bien = v.id_bien
10 and b.Type_local='Appartement'
11 and b.Total_pieces>4
12 group by r.nom_region
13 order by avg(v.valeur)/avg(b.Surface_car) desc
14
```

The results are displayed in a table with 15 rows and 2 columns:

	nom region	prix_metre2
1	LA REUNION	9957.65
2	ILE-DE-FRANCE	3859.37
3	PROVENCE-ALPES-COTE D'AZUR	2238.12
4	CORSE	2088.16
5	AUVERGNE-RHONE-ALPES	1723.99
6	NOUVELLE-AQUITAINE	1488.74
7	BRETAGNE	1339.19
8	HAUTS-DE-FRANCE	1237.14
9	CENTRE-VAL DE LOIRE	1136.86
10	PAYS DE LA LOIRE	1130.18
11	OCCITANIE	1104.18
12	GRAND EST	1050.55
13	NORMANDIE	999.55
14	BOURGOGNE-FRANCHE-COMTE	745.56
15	MARTINIQUE	447.59

The status bar at the bottom indicates an error: "[12:10:40] Erreur pendant l'exécution de la requête sur la base de données « db30 » : near 'tab2': syntax error". The query execution time is 0.383 seconds.

--9 Liste des communes ayant eu au moins 50 ventes au 1er trimestre

select

c.Nom_commune,

count(v.id_vente) as nombre_de_ventes

from Region r, Departement d, Commune c, Bien b, Vente v

where r.id_code_region = d.id_code_region

and d.id_code_departement = c.id_code_departement

and c.id_code_dep_commune = b.id_code_dep_commune

and b.id_bien = v.id_bien

and v.date >='2020-01-01' and v.date <'2020-04-01'

group by c.Nom_commune

having count(v.id_vente)>=50;

	Nom commune	nombre de ventes
1	AJACCIO	55
2	ANGERS	64
3	ANTIBES	76
4	ASNIERES-SUR-SEINE	82
5	BORDEAUX	157
6	BOULOGNE-BILLANCOURT	99
7	COURBEVOIE	76
8	GRENOBLE	110
9	LA CIOTAT	64
10	LEVALLOIS-PERRET	60
11	LILLE	72
12	MARSEILLE 1ER ARRONDISSEMENT	72
13	MARSEILLE 4E ARRONDISSEMENT	72
14	MARSEILLE 8E ARRONDISSEMENT	80
15	MARSEILLE 9E ARRONDISSEMENT	64
16	MONTREUIL	68
17	NANTES	123
18	NICE	174
19	NIMES	64
20	PARIS 10E ARRONDISSEMENT	115
21	PARIS 11E ARRONDISSEMENT	171
22	PARIS 12E ARRONDISSEMENT	109
23	PARIS 13E ARRONDISSEMENT	94
24	PARIS 14E ARRONDISSEMENT	146
25	PARIS 15E ARRONDISSEMENT	215

SQLStudio (3.4.0) - [Éditeur SQL 1]

Database Structure View Tools Help

Base de données

Filtre par nom

db3 (SQLite 3)

db3_1 (SQLite 3)

Tables (6)

- Bel
- Colomes (12)
- Index
- Déclencheurs
- Commune
- Département
- Région
- Vente
- Vente_2
- Vues

Requête Résultats Historique

Table Formulaire

Nombre de lignes chargées : 46

	Nom commune	nombre de ventes
22	PARIS 13E ARRONDISSEMENT	94
23	PARIS 14E ARRONDISSEMENT	146
24	PARIS 15E ARRONDISSEMENT	215
25	PARIS 16E ARRONDISSEMENT	165
26	PARIS 17E ARRONDISSEMENT	228
27	PARIS 18E ARRONDISSEMENT	209
28	PARIS 19E ARRONDISSEMENT	105
29	PARIS 20E ARRONDISSEMENT	138
30	PARIS 3E ARRONDISSEMENT	79
31	PARIS 4E ARRONDISSEMENT	51
32	PARIS 5E ARRONDISSEMENT	73
33	PARIS 6E ARRONDISSEMENT	89
34	PARIS 7E ARRONDISSEMENT	91
35	PARIS 8E ARRONDISSEMENT	59
36	PARIS 9E ARRONDISSEMENT	100
37	PUTEAUX	57
38	RENNES	65
39	RUEIL-MALMAISON	69
40	SAINT-MAUR-DES-FOSSES	56
41	SETE	61
42	TOULON	64
43	TOULOUSE	77
44	VERSAILLES	58
45	VINCENNES	68

Barre d'état

[20:00:27] Requête terminée en 0.405 seconde(s).

[20:16:04] Requête terminée en 0.230 seconde(s).

[20:17:59] Requête terminée en 0.258 seconde(s).

SQL editor 1 Éditeur SQL 1

Taper ici pour rechercher

10°C Nuageux FRA 20:19 09/01/2023

--10 Différence en pourcentage du prix au mètre carré entre un appartement de 2 pièces et un appartement de 3 pièces

```
with tab1 as (  
select  
b.Total_pieces,  
avg(v.valeur)/avg(b.Surface_car) as prix1  
from Region r, Departement d, Commune c, Bien b, Vente v  
where r.id_code_region = d.id_code_region  
and d.id_code_departement = c.id_code_departement  
and c.id_code_dep_commune = b.id_code_dep_commune  
and b.id_bien = v.id_bien  
and b.Type_local='Appartement'  
group by b.Total_pieces  
having b.Total_pieces=2),  
tab2 as (select  
b.Total_pieces,  
avg(v.valeur)/avg(b.Surface_car) as prix2  
from Region r, Departement d, Commune c, Bien b, Vente v  
where r.id_code_region = d.id_code_region  
and d.id_code_departement = c.id_code_departement  
and c.id_code_dep_commune = b.id_code_dep_commune  
and b.id_bien = v.id_bien  
and b.Type_local='Appartement'  
group by b.Total_pieces  
having b.Total_pieces=3)  
select  
round(tab1.prix1, 2) as prix_2_pieces,  
round(tab2.prix2,2) as prix_3_pieces,  
100-round(tab2.prix2*100/tab1.prix1, 2) as difference_pourcentage  
from tab1,tab2;
```

SQLStudio (3.4.0) - [SQL editor 1]

Database Structure View Tools Help

Base de données

Requête Historique

1 --10 Différence en pourcentage du prix au mètre carré entre un appartement de 2 pièces et un appartement de 3 pièces

```
2 with tab1 as (
3 select
4 b.Total_pieces,
5 avg(v.valeur)/avg(b.Surface_car) as prix1
6 from Region r, Departement d, Commune c, Bien b, Vente v
7 where r.id_code_region = d.id_code_region
8 and d.id_code_departement = c.id_code_departement
9 and c.id_code_des_commune = b.id_code_des_commune
10 and b.id_bien = v.id_bien
11 and b.type_local='Appartement'
12 group by b.Total_pieces
13 having b.Total_pieces=2),
14 tab2 as (select
15 b.Total_pieces,
16 avg(v.valeur)/avg(b.Surface_car) as prix2
17 from Region r, Departement d, Commune c, Bien b, Vente v
18 where r.id_code_region = d.id_code_region
19 and d.id_code_departement = c.id_code_departement
20 and c.id_code_des_commune = b.id_code_des_commune
21 and b.id_bien = v.id_bien
22 and b.type_local='Appartement'
23 group by b.Total_pieces
24 having b.Total_pieces=3)
25 select
26 round(tab1.prix1, 2) as prix_2_pieces,
27 round(tab1.prix1, 2) as prix_3_pieces,
28 100-round(tab2.prix2*100/tab1.prix1, 2) as difference_pourcentage
29 from tab1, tab2;
30 ]
```

Table Formulaire

Nombre de lignes chargées : 1

	prix 2 pieces	prix 3 pieces	difference pourcentage
1	6004.61	3796.85	36.77

Barre d'état

[11:58:29] Requête terminée en 4.117 seconde(s).

[12:00:30] Requête terminée en 0.873 seconde(s).

[12:05:41] Requête terminée en 0.512 seconde(s).

[12:06:25] Requête terminée en 0.328 seconde(s).

SQL editor 1

Taper ici pour rechercher

10°C Nuageux FRA 12:08 13/01/2023

```
--11 Les moyennes de valeurs foncières pour le top 3 des communes des département 6,
13, 33, 59, 69
with tab_1 as (select
d.id_code_departement,
c.Nom_commune,
round(avg(v.valeur),0) as moyenne_de_vf,
rank() over(partition by d.id_code_departement order by avg(v.valeur) desc ) as rank
from Region r, Departement d, Commune c, Bien b, Vente v
where r.id_code_region = d.id_code_region
and d.id_code_departement = c.id_code_departement
and c.id_code_dep_commune = b.id_code_dep_commune
and b.id_bien = v.id_bien
group by d.id_code_departement, c.Nom_commune
having d.id_code_departement in ('06', '13', '33', '59', '69')
order by d.id_code_departement, avg(v.valeur) desc)
select * from tab_1 where rank<=3;
```

SQLStudio 3.4.0 - [Éditeur SQL 1]

Requête

```
1 --11 Les moyennes de valeurs foncières pour le top 3 des communes des département 6, 13, 33, 59, 69
2 with tab_1 as (select
3 d.id_code_departement,
4 c.Nom_commune,
5 round(avg(v.valeur),0) as moyenne_de_vf,
6 rank() over(partition by d.id_code_departement order by avg(v.valeur) desc ) as rank
7 from Region r, Departement d, Commune c, Bien b, Vente v
8 where r.id_code_region = d.id_code_region
9 and d.id_code_departement = c.id_code_departement
10 and c.id_code_dep_commune = b.id_code_dep_commune
11 and b.id_bien = v.id_bien
12 group by d.id_code_departement, c.Nom_commune
13 having d.id_code_departement in ('06', '13', '33', '59', '69')
14 order by d.id_code_departement, avg(v.valeur) desc)
15 select * from tab_1 where rank<=3;
```

Table

id code departement	Nom commune	moyenne de vf	rank
1 06	L'ESCARPINE	720540	1
2 06	ROQUEBILLIERE	627000	2
3 06	LA BOLLÈNE-VESURIE	395000	3
4 13	CHATEAUNEUF-LES-MARTIGUES	330000	1
5 13	LA BOUILLADISSE	268545	2
6 13	ROQUEVAIRE	257500	3
7 33	LANGOIRAN	555000	1
8 33	MARCHEPRIME	524036	2
9 33	PAILLET	403333	3
10 59	HASNON	565000	1
11 59	FACHES-THUMESNIL	271950	2
12 59	WAMBRECHIES	224750	3
13 69	SAINT-GERMAIN-AU-MONT-D'OR	932400	1
14 69	SAINT-GERMAIN-NUELLES	872850	2
15 69	MORNANT	534000	3

Barre d'état

[20:23:08] Requête terminée en 0.510 seconde(s).

[20:23:45] Requête terminée en 0.495 seconde(s).

--12 Les 20 communes avec le plus de transactions pour 1000 habitants pour les communes qui dépassent les 10000 habitants

select

c.Nom_commune,

round(count(v.id_vente)*1000.0/c.Population_totale, 2) as trans_par_1000hbtms

from Region r, Departement d, Commune c, Bien b, Vente v

where r.id_code_region = d.id_code_region

and d.id_code_departement = c.id_code_departement

and c.id_code_dep_commune = b.id_code_dep_commune

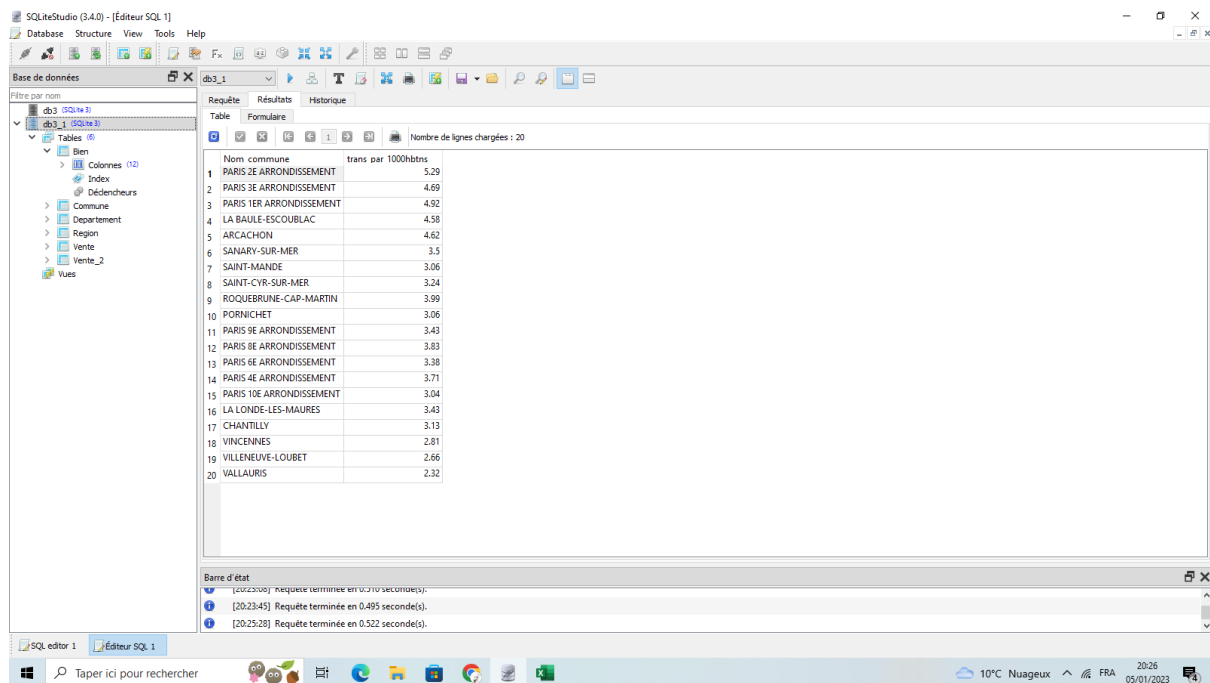
and b.id_bien = v.id_bien

and c.Population_totale>10000

group by c.Nom_commune

order by count(v.id_vente)*1000/c.Population_totale desc

limit 20;



SQLStudio 3.4.0 - [Éditeur SQL 1]

Base de données: db3_1 (SQLite 3)

Requête: trans par 1000hbtms

Résultats: 20 lignes chargées

Nom commune	trans par 1000hbtms
PARIS 3E ARRONDISSEMENT	5.29
PARIS 3E ARRONDISSEMENT	4.69
PARIS 1ER ARRONDISSEMENT	4.92
LA BAULE-ESCOUBLAC	4.58
ARCACHON	4.62
SANARY-SUR-MER	3.5
SAINT-MANDE	3.06
SAINT-CYR-SUR-MER	3.24
ROQUEBRUNE-CAP-MARTIN	3.99
PORNICHE	3.06
PARIS 9E ARRONDISSEMENT	3.43
PARIS 8E ARRONDISSEMENT	3.83
PARIS 6E ARRONDISSEMENT	3.38
PARIS 4E ARRONDISSEMENT	3.71
PARIS 10E ARRONDISSEMENT	3.04
LA LONDE-LES-MAURES	3.43
CHANTILLY	3.13
VINCENNES	2.81
VILLENEUVE-LOUBET	2.66
VALLAURIS	2.32

Barre d'état: [20:23:06] Requête terminée en 0.370 seconde(s).
[20:23:45] Requête terminée en 0.495 seconde(s).
[20:25:28] Requête terminée en 0.522 seconde(s).