

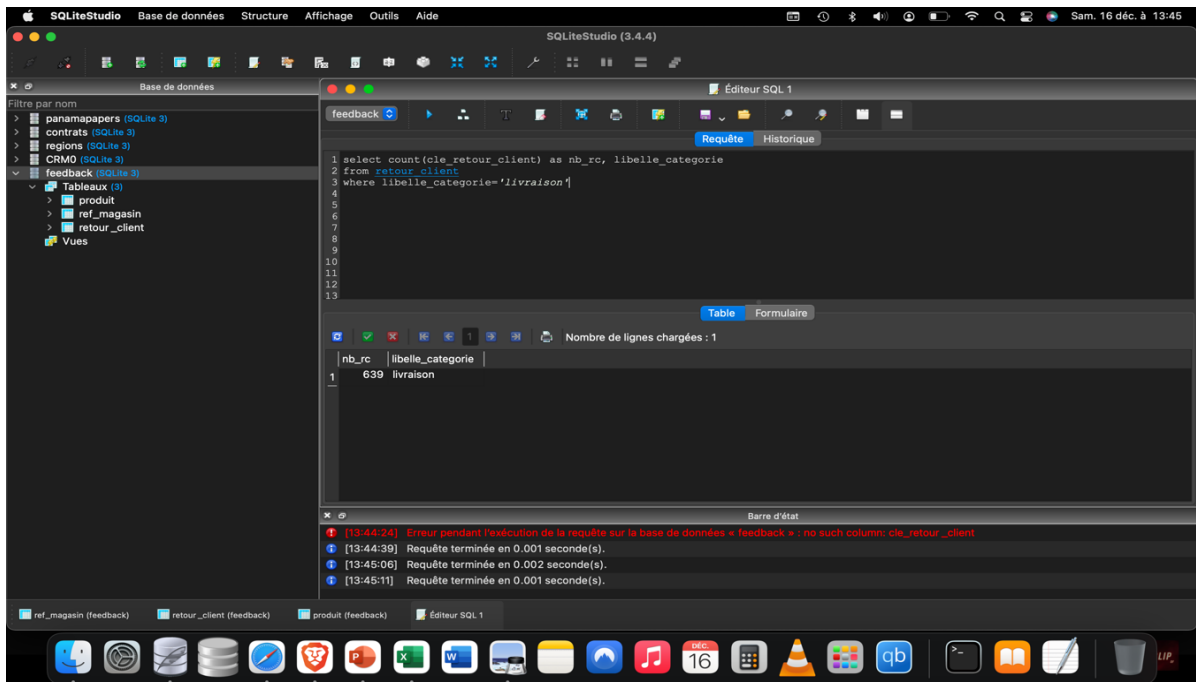
## LISTE DES REQUÊTES

### Projet 5 : interrogez une base de données SQL pour suivre la satisfaction client

Auteur : Sofiane AYOUB

Date : 11/12/2023

#### Requête 1 : quel est le nombre de retours clients sur la livraison ?



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

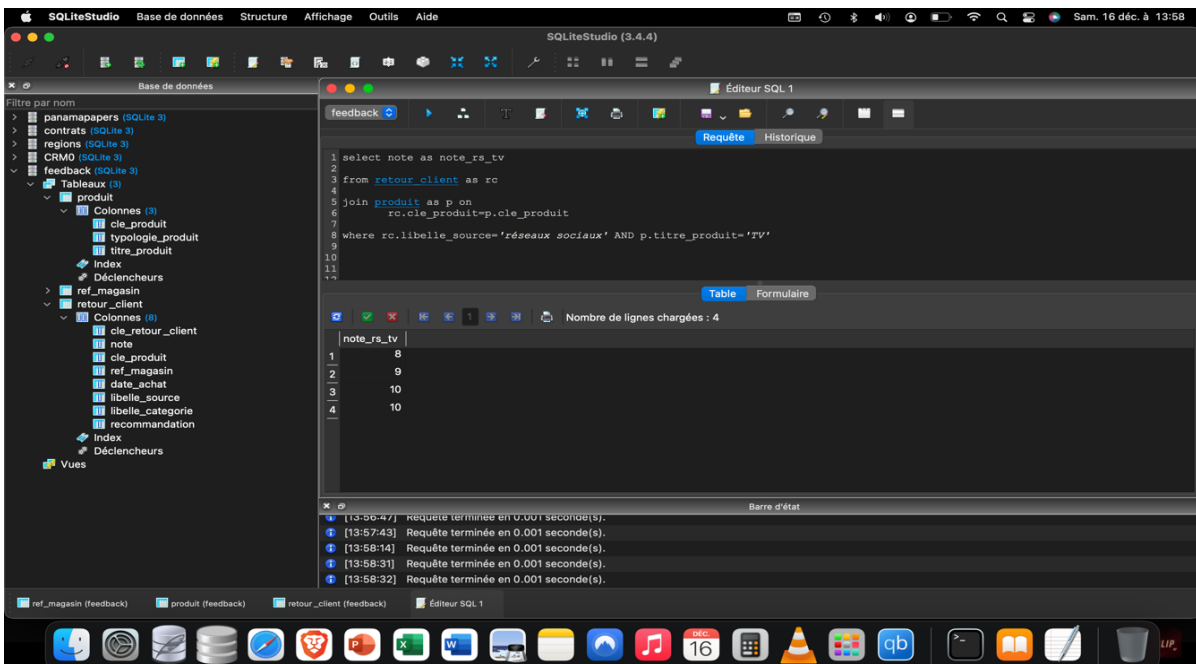
```
1 select count(cle_retour_client) as nb_rc, libelle_categorie
2 from retour_client
3 where libelle_categorie='livraison'
```

The result table shows one row:

nb_rc	libelle_categorie
639	livraison

The status bar at the bottom indicates the query was executed successfully at 13:44:24.

#### Requête 2 : quelle est la liste des notes des clients sur les réseaux sociaux sur les TV ?



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

```
1 select note as note_rs_tv
2
3 from retour_client as rc
4
5 join produit as p on
6 rc.cle_produit=p.cle_produit
7
8 where rc.libelle_source='réseaux sociaux' AND p.titre_produit='TV'
```

The result table shows four rows:

note_rs_tv
8
9
10
10

The status bar at the bottom indicates the query was executed successfully at 13:56:47.

**Requête 3** : quelle est la note moyenne pour chaque catégorie de produit ? (Classé de la meilleure à la moins bonne).

The screenshot shows the SQLiteStudio interface. The left sidebar displays the database structure with the 'feedback' table selected. The central SQL editor contains the following query:

```
1 select typologie_produit, round(avg(note), 2) as avg_note
2
3 from produit as p
4
5 join retour_client as rc on
6   p.cle_produit=rc.cle_produit
7
8 group by typologie_produit
9
10 order by avg_note desc
11
12
13
14
15
16
17
```

The 'Requête' tab is active, showing the results in a table with 4 rows:

typologie_produit	avg_note
1 High-Tech	8.16
2 Loisirs	8.09
3 Alimentaire	8.04
4 Maison	7.85

The status bar at the bottom indicates 'Nombre de lignes chargées : 4'.

**Requête 4** : quels sont les 5 magasins avec les meilleurs notes moyennes ?

The screenshot shows the SQLiteStudio interface. The left sidebar displays the database structure with the 'feedback' table selected. The central SQL editor contains the following query:

```
1 select retour_client.ref_magasin, round(avg(note), 2) as avg_note
2
3 from retour_client
4
5 group by retour_client.ref_magasin
6
7 order by avg_note desc
8
9 limit 5
10
11
12
13
14
15
16
17
```

The 'Requête' tab is active, showing the results in a table with 5 rows:

ref_magasin	avg_note
1 75	8.73
2 78	8.55
3 62	8.5
4 23	8.48
5 19	8.45

The status bar at the bottom indicates 'Nombre de lignes chargées : 5'.

### Requête 5 : quels sont les magasins qui ont plus de 12 feedbacks sur le drive ?

The screenshot shows the SQLiteStudio interface with the following SQL query in the editor:

```
1 select ref_magasin,count(*) as nbr_retour_drive
2
3 from retour_client
4
5 where libelle_categorie='drive'
6
7 group by ref_magasin
8
9 having count(*)>=12
10
11 order by nbr_retour_drive
12
13
```

The results table shows 4 rows:

ref_magasin	nbr_retour_drive
57	12
45	13
63	13
67	14

The status bar at the bottom indicates "Nombre de lignes chargées : 4".

### Requête 6 : quel est le classement des départements par note ?

The screenshot shows the SQLiteStudio interface with the following SQL query in the editor:

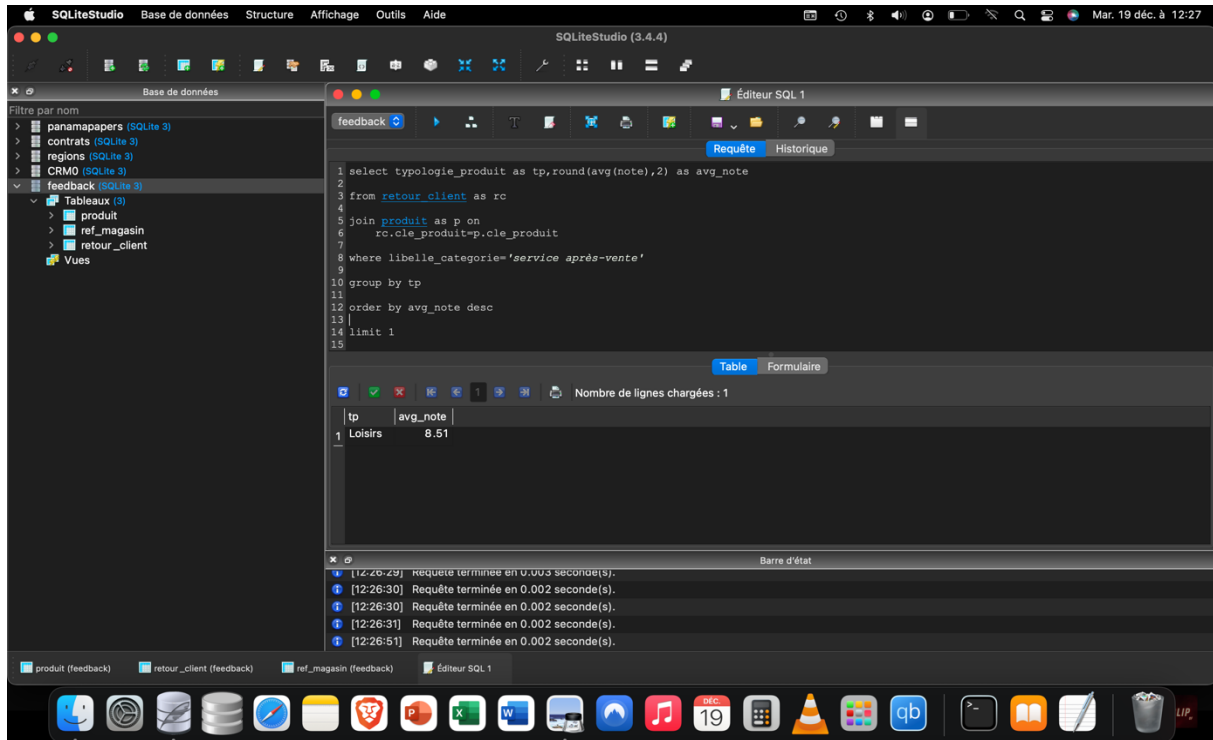
```
1 select departement, round(avg(note),2) as avg_note
2
3 from retour_client as rc
4
5 join ref_magasin as rm on
6 rc.ref_magasin=rm.ref_magasin
7
8 group by departement
9
10 order by avg_note desc
11
12
```

The results table shows 8 rows:

departement	avg_note
95	8.14
75	8.11
94	8.06
91	8.05
77	8.04
92	8.03
78	8.02
93	7.94

The status bar at the bottom indicates "Nombre de lignes chargées : 8".

## Requête 7 : quelle est la typologie de produit qui apporte le meilleur service après-vente ?



The screenshot shows the SQLiteStudio interface with the following SQL query in the 'Éditeur SQL 1' window:

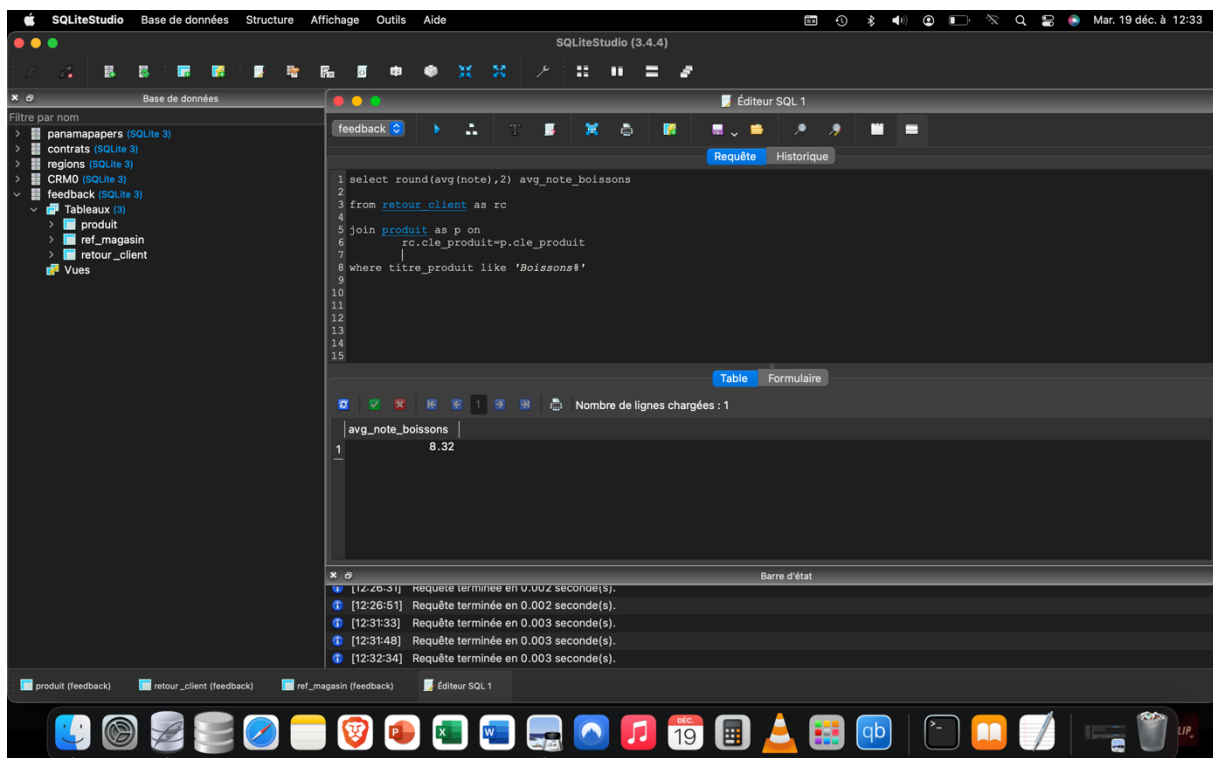
```
1 select typologie_produit as tp, round(avg(note),2) as avg_note
2
3 from retour_client as rc
4
5 join produit as p on
6   rc.cle_produit=p.cle_produit
7
8 where libelle_categorie='service après-vente'
9
10 group by tp
11
12 order by avg_note desc
13
14 limit 1
15
```

The 'Table' tab shows the result:

tp	avg_note
1 Loisirs	8.51

The status bar at the bottom indicates the query was terminated in 0.002 seconds.

## Requête 8 : quelle est la note moyenne sur l'ensemble des boissons ?



The screenshot shows the SQLiteStudio interface with the following SQL query in the 'Éditeur SQL 1' window:

```
1 select round(avg(note),2) avg_note_boissons
2
3 from retour_client as rc
4
5 join produit as p on
6   rc.cle_produit=p.cle_produit
7
8 where titre_produit like 'Boissons%'
9
10
11
12
13
14
15
```

The 'Table' tab shows the result:

avg_note_boissons
1 8.32

The status bar at the bottom indicates the query was terminated in 0.002 seconds.

**Requête 9 : quel est le classement des jours de la semaine où l'expérience client est la meilleure expérience en magasin ?**

The screenshot shows the SQLiteStudio interface. The left sidebar displays the database structure with 'feedback' selected. The main editor contains the following SQL query:

```
1 select strftime('%w',date_achat) as jour_semaine,round(avg(note),2) as avg_note_experience_magasin
2
3 from retour_client as rc
4
5 where libelle_categorie='expérience en magasin'
6
7 group by strftime('%w',date_achat)
8
9 order by avg_note_experience_magasin desc
10
11
12
13
```

The results are displayed in a table with 7 rows:

jour_semaine	avg_note_experience_magasin
6	8.34
0	8.18
5	8.07
4	8.04
3	7.99
2	7.95
1	7.74

The status bar at the bottom indicates 'Nombre de lignes chargées : 7'.

**Requête 10 : sur quel mois a-t-on le plus de retour sur le service après-vente ?**

The screenshot shows the SQLiteStudio interface. The left sidebar displays the database structure with 'feedback' selected. The main editor contains the following SQL query:

```
1 select strftime('%m',date_achat) as mois,libelle_categorie,count(*) as nb_retour_sav
2
3 from retour_client as rc
4
5 where libelle_categorie='service après-vente'
6
7 group by strftime('%m',date_achat)
8
9 order by nb_retour_sav desc
10
11 limit 1
12
13
```

The results are displayed in a table with 1 row:

mois	libelle_categorie	nb_retour_sav
10	service après-vente	55

The status bar at the bottom indicates 'Nombre de lignes chargées : 1'.

**Requête 11** : quel est le pourcentage de recommandations clients ? (Comptabiliser le nombre de retour client qui ont répondu « Oui » divisé par le nombre de retours total)

The screenshot shows the SQLiteStudio interface. The left sidebar displays the database structure with the 'feedback' table selected. The main editor contains the following SQL query:

```
1 select reco_positif * 100/total_retour,total_retour
2 from(
3
4     select sum(case when recommandation is true then 1 else 0 end) as reco_positif,
5           sum(case when recommandation in (true,false) then 1 else 0 end) as total_retour
6           from retour_client)
7
8
9
10
11
12
13
```

The query results are displayed in a table with 2 columns: 'reco\_positif \* 100 / total\_retour' and 'total\_retour'. The results show a value of 90 for the percentage and 2326 for the total number of returns.

reco_positif * 100 / total_retour	total_retour
90	2326

The status bar at the bottom indicates that the query was terminated in 0.002 seconds.

**Requête 12** : quels sont les magasins qui ont une note inférieure à la moyenne ?

The screenshot shows the SQLiteStudio interface. The left sidebar displays the database structure with the 'feedback' table selected. The main editor contains the following SQL query:

```
1 select ref_magasin, round(avg(note),2) as avg_note_mag, avg_global.avg_note
2
3 from (select round(avg(note),2) as avg_note from retour_client) as avg_global, retour_client rc
4
5 group by ref_magasin
6
7 having avg(note) <= avg_global.avg_note
8
9 order by avg_note_mag asc
10
11
```

The query results are displayed in a table with 3 columns: 'ref\_magasin', 'avg\_note\_mag', and 'avg\_note'. The results show 10 rows of data, sorted by the average rating of each store.

ref_magasin	avg_note_mag	avg_note
60	7.38	8.05
81	7.44	8.05
82	7.53	8.05
46	7.56	8.05
55	7.59	8.05
24	7.62	8.05
80	7.62	8.05
8	7.66	8.05
44	7.67	8.05
74	7.7	8.05

The status bar at the bottom indicates that the query was terminated in 0.003 seconds.

### Requête 13 : quelles sont les typologies produit qui ont amélioré leur moyenne entre le 1<sup>er</sup> et le 2<sup>ème</sup> trimestre 2021 ?

The screenshot shows the SQLiteStudio interface with a SQL query in the 'Éditeur SQL 1' window. The query compares the average product typology scores for the first and second quarters of 2021. The results are displayed in a table below the query editor.

```
1 SELECT Trimestre1.typologie_produit,
2       Trimestre1.avg_typologie_produit as avg_T1,
3       Trimestre2.avg_typologie_produit as avg_T2
4 FROM( select typologie_produit, round(avg(note),2) as avg_typologie_produit
5       from retour_client as rc
6       left join produit p
7       on rc.cle_produit = p.cle_produit
8       where strftime('%m',date_achat) in ('01','02','03')
9
10      GROUP BY typologie_produit) Trimestre1,
11
12 ( select typologie_produit, round(avg(note),2) as avg_typologie_produit from retour_client as rc
13       left join produit p
14       on rc.cle_produit = p.cle_produit
15       where strftime('%m',date_achat) in ('04','05','06')
16      GROUP BY typologie_produit) Trimestre2
17 WHERE Trimestre2.typologie_produit = Trimestre1.typologie_produit
18 AND Trimestre2.avg_typologie_produit > Trimestre1.avg_typologie_produit
```

typologie_produit	avg_T1	avg_T2
1 Alimentaire	7.99	8.06
2 Loisirs	8	8.34

Barre d'état: Requête terminée en 0.009 seconde(s).

### Requête Net Promoter Score : mesure du NPS

The screenshot shows the SQLiteStudio interface with a SQL query in the 'Éditeur SQL 1' window. The query calculates the Net Promoter Score (NPS) based on customer feedback. The results are displayed in a table below the query editor.

```
1 SELECT promoteur*100/total - detracteur*100/total as score_nps
2
3 FROM(select sum(case when note > 8 then 1 else 0 end) as promoteur,
4          sum(case when note <= 6 then 1 else 0 end) as detracteur,
5          count(note) as total from retour_client)
6
7
8
9
10
11
12
```

score_nps
31

Barre d'état: Requête terminée en 0.002 seconde(s).