



# BestMarket

*Manipulez une base de données avec SQL pour suivre la satisfaction client*

*Felizarda*

*Lando*

*20/12/2024*

# 1) Contexte et expression du besoin



*Le service client de BestMarket dirigé par Olivier a besoin d'aide sur le customer data feedback (demande retour client).*

*Au quotidiens, l'entreprise est sollicité par les consommateurs pour des demandes diverses (réclamations, informations, suggestions) avec des exigences toujours plus importantes.*

*Les données provenant de sources diverses (réseaux sociaux, campagne d'emailing, application mobil, feedback téléphonique après contact) les équipes ont du mal à s'y retrouver.*

*Oliver fait appel à nous pour synthétiser les données et répondre aux questions que se pose régulièrement son équipe.*

*Les réponses devront permettre de comprendre comment l'expérience client doit être améliorée car, c'est un enjeu très important qui permettra à BestMarket de fidéliser plus de clients.*

## 2) Sauvegarde et stockage de la BDD




- Rappel des bonnes pratiques en termes de stratégie de sauvegarde, de stockage et d'accès aux données de notre base de données
- **Effectuer des sauvegardes fréquentes des données**, que celles-ci soient sous forme papier ou électronique. Il peut être opportun de prévoir des sauvegardes des sauvegardes incrémentales quotidiennes et des sauvegardes complètes à intervalles réguliers.
- **Stocker au moins une sauvegarde sur un site géographiquement distinct** du site d'exploitation.
- Isoler au moins une sauvegarde hors ligne, déconnectée du réseau de l'entreprise.
- **Protéger les données sauvegardées au même niveau de sécurité que celles stockées sur les serveurs d'exploitation.**

### 3) Méthodologie suivie

1- Dans un premier temps, **complétude du dictionnaire des données**

	Nom du champs	Type de données	Taille	Contrainte	Description
Table Retour client	cle_retour_client	INT		Clé primaire	ID unique pour les retours clients
	note	INT			Note donnée par le client, comprise entre 0 et 10, la note est la réponse à la question : "Sur une échelle de 0 à 10 quelle est la probabilité que vous recommandiez notre ent
	Cle_produit	INT		Clé secondaire	ID des produits
	ref_magasin	INT		Clé secondaire	ID des magasins
	date_achat	DATE			Date à laquelle l'achat du client a eu lieu
	libelle_source	CHAR	50		Libellé de la source d'où provient le retour client (Réseaux sociaux, téléphone, email)
	libelle_categorie	CHAR	50		Libellé de la catégorie du retour client (Drive, service après-vente, qualité produit, expérience en magasin, livraison)
	recommandation	CHAR			Recommandation laissée par le client à la question 'Recommandez vous l'entreprise?' True / False
Table Produit	cle_produit	INT		Clé primaire	ID unique pour les produits
	titre_produit	CHAR	50		Libellé des produits
	typologie_produit	INT			Typologie des produits (Alimentaire, High-tech etc...)
Table Ref_magasin	cle-ref-magasin	INT		Clé primaire	ID unique pour les références en magasin
	departement	INT	5		code postale
	libelle_de_commune	CHAR	30		nom des communes
	population	INT			nombre habitants de la commune
	geo_point_2d	VARCHAR			localisation géographique

2- Création de la BDD 'customer\_data\_feedback' avec la surface graphique

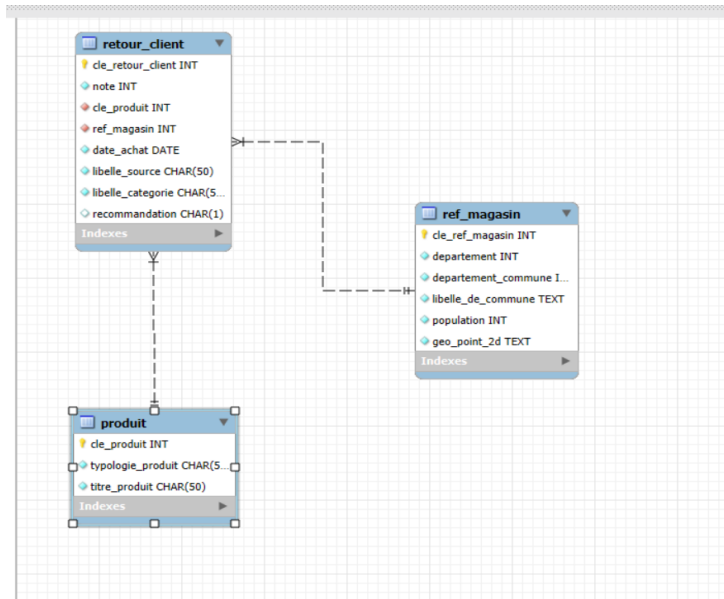
 Name:  Specify the name of the schema here

Refactor model, changing all references

Charset/Collation:   The character set and its collation are

```
1 ALTER SCHEMA 'customerdatafeedback' DEFAULT COLLATE utf8mb4_bin ;
2
```

## Méthodologie suivie suite



3- Après avoir compléter le dictionnaire des données, création du nouveau **schéma relationnel**

The screenshot shows the 'Table Name' field set to `ref_magasin` in the `customer_data_feedback` schema. The 'Columns' tab is active, displaying the following columns:

Column Name	Datatype	PK	IN	UN	B	UN	2P	AI	G	Default/Expression
ref_magasin	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
departement	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
departement_commune	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
libelle_de_commune	TEXT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
population	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
geo_point_2d	TEXT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

The 'Columns' tab is selected, and the 'Apply' button is visible at the bottom right.

```
1 CREATE TABLE `customerdatafeedback`.`ref_magasin` (  
2   `ref_magasin` INT NOT NULL,  
3   `departement` INT NOT NULL,  
4   `departement` INT NOT NULL,  
5   `libelle_de_commune` VARCHAR(50) NOT NULL,  
6   `goe_point` VARCHAR(50) NULL,  
7   `population` INT NOT NULL,  
8   PRIMARY KEY (`ref_magasin`));  
9
```

4-Création des tables et des **foreign key**, clé produit et clé\_ref\_magasin

## Méthodologie suivie suite

### 4- Creation des TABLES produit, ref\_magasin, et retour\_client , des clefs primaires et clés étrangères.

Table Name: retour\_client Schema: customer\_data\_feedback  
Charset/Collation: utf8mb4 Engine: InnoDB

Comments:

Column Name	Datatype	PK	NU	UQ	B	UN	2P	AI	G	DefaultExpression
cle_retour_client	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
note	INT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
cle_produit	INT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
ref_magasin	INT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
date_schet	DATE	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
libelle_source	CHAR(30)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
libelle_categorie	CHAR(30)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
recommandation	CHAR(30)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL

Column Name: Data Type: Default: Storage: ☐ Virtual ☐ Stored ☐ Primary Key ☐ Not Null ☐ Unique ☐ Binary ☐ Unsigned ☐ Zero Fill ☐ Auto Increment ☐ Generated

Columns Indices Foreign Keys Triggers Partitioning Options

Apply Revert

Table Name: produit Schema: customer\_data\_feedback  
Charset/Collation: utf8mb4 Engine: InnoDB

Comments:

Column Name	Datatype	PK	NU	UQ	B	UN	2P	AI	G	DefaultExpression
cle_produit	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
nom_magasin_produit	CHAR(30)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
titre_produit	CHAR(30)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Column Name: Data Type: Default: Storage: ☐ Virtual ☐ Stored ☐ Primary Key ☐ Not Null ☐ Unique ☐ Binary ☐ Unsigned ☐ Zero Fill ☐ Auto Increment ☐ Generated

Columns Indices Foreign Keys Triggers Partitioning Options

Table Name: retour\_client Schema: customer\_data\_feedback  
Charset/Collation: utf8mb4 Engine: InnoDB

Comments:

Foreign Key Name	Referenced Table	Column	Referenced Column	Foreign Key Options
cle_produit	customer_data_feedback.'produit'	<input type="checkbox"/> cle_retour_client		On Update: RESTRICT
cle_ref_magasin	customer_data_feedback.'ref_magasin'	<input type="checkbox"/> note		On Delete: RESTRICT
		<input checked="" type="checkbox"/> cle_produit	cle_produit	<input type="checkbox"/> Skip in SQL generation
		<input type="checkbox"/> ref_magasin		
		<input type="checkbox"/> date_schet		
		<input type="checkbox"/> libelle_source		
		<input type="checkbox"/> libelle_categorie		
		<input type="checkbox"/> recommandation		

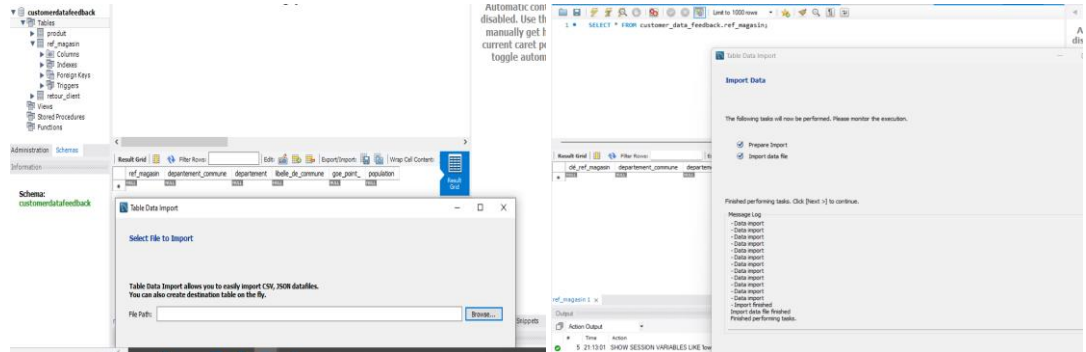
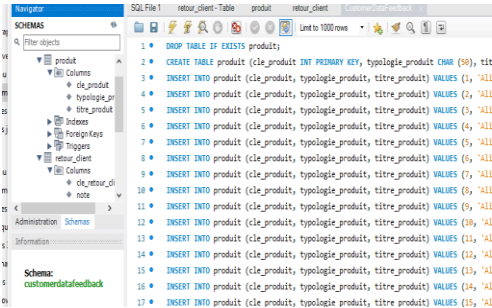
Foreign Key Comment:

Columns Indices Foreign Keys Triggers Partitioning Options

Apply Revert

## Méthodologie suivie suite

- Insertion des données dans les tables**



5- Une fois les tables créées **exécution de la commande SQL** pour remplir les colonnes des tables *produit* et *return\_client*

Pour la table *ref\_magasin* **'table data import'** pour importer les données du fichier csv.

## 4) Requêtes SQL et Analyses

The screenshot shows a SQL query editor with the following code:

```
1 -- requete 1:Le nombre de retours clients sur la livraison
2
3 • SELECT count(*) as 'nombre de retour client sur la livraison'
4   from retour_client
5   where libelle_categorie like '%livraison';
```

Below the code, the 'Result Grid' shows the following data:

nombre de retour client sur la livraison
639

Il y a **639** retours client sur la livraison

The screenshot shows a SQL query editor with the following code:

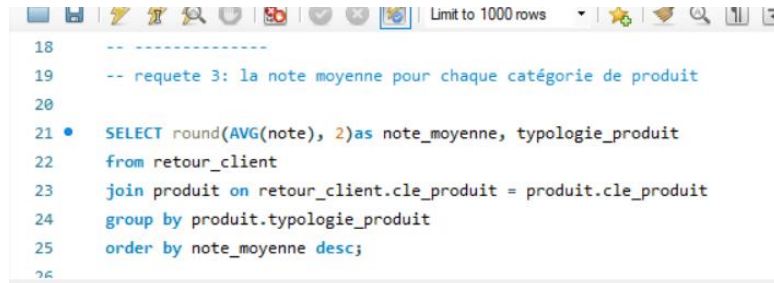
```
14 -- requete 2 La liste des notes des clients sur les réseaux sociaux sur les TV
15
16 • SELECT note as 'note client réseaux sociaux'
17   from retour_client
18   join produit on retour_client.cle_produit = produit.cle_produit
19   where produit.titre_produit = 'TV' and libelle_source = 'réseaux sociaux'
20   group by note;
```

Below the code, the 'Result Grid' shows the following data:

note client réseaux sociaux
10
9
8

**Les notes** attribuées au **TV** sur les réseaux sociaux sont: **8, 9 et 10**





```

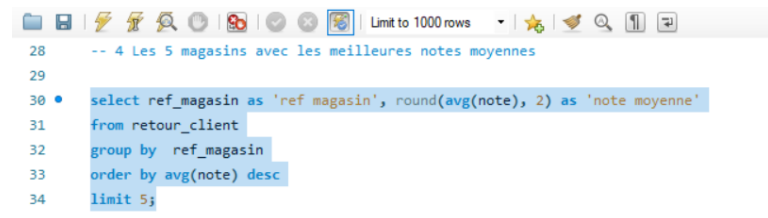
18  -----
19  -- requete 3: la note moyenne pour chaque catégorie de produit
20
21  • SELECT round(AVG(note), 2) as note_moyenne, typologie_produit
22    from retour_client
23   join produit on retour_client.cle_produit = produit.cle_produit
24   group by produit.typologie_produit
25   order by note_moyenne desc;
26

```

Result Grid

note_moyenne	typologie_produit
8.16	High-Tech
8.09	Loisirs
8.04	Alimentaire
7.85	Maison

Les notes moyenne de chaque catégorie de produit sont très proches. La meilleure moyenne revient aux produits **High-Tech 8,16** et la moins bonne aux produits **Maison 7,85**



```

28  -- 4 Les 5 magasins avec les meilleures notes moyennes
29
30  • select ref_magasin as 'ref magasin', round(avg(note), 2) as 'note moyenne'
31    from retour_client
32   group by ref_magasin
33   order by avg(note) desc
34   limit 5;
35

```

Result Grid

ref magasin	note moyenne
75	8.73
78	8.55
62	8.50
23	8.48
19	8.45

Les 5 magasins avec la meilleure moyenne sont le **magasin 75;78;62;23 et 19**

```

38  -- - requete 5: Les magasins qui ont plus de 12 feedbacks sur le drive
39
40  • select ref_magasin , count(recommandation)as 'nombre de feedback'
41  from retour_client
42  where libelle_categorie = 'drive'
43  group by ref_magasin
44  having count(recommandation)>12
45  order by ref_magasin;
46
47

```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
ref_magasin	nombre de feedback		
45	13		
63	13		
67	14		

Ce sont les magasins 45; 63 et 67 qui ont plus de 12 feedbacks

```

49  -- requete 6 Le classement des départements par note
50
51
52  • select departement, round(avg(note), 2) as note_moyenne_dep
53  from retour_client
54  join ref_magasin on retour_client.ref_magasin = ref_magasin.cle_ref_magasin
55  group by departement
56  order by note_moyenne_dep desc;
57
58
59

```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
departement	note_moyenne_dep		
95	8.14		
75	8.11		
94	8.06		
91	8.05		
77	8.04		
92	8.03		
78	8.02		
93	7.94		

Dans le classement des départements on constate que le département qui a la **meilleure note est le 95 avec 8,14** et celui qui a la **moins bonne le 93 avec 7,94**

```

62 -- requete 7: La typologie de produit qui apporte le meilleur service après-vente
63
64 • select typologie_produit as produit, round(avg(note), 2) as `meilleure service après vente`
65 from retour_client
66 join produit on retour_client.cle_produit = produit.cle_produit
67 where libelle_categorie = 'service après-vente'
68 group by produit.typologie_produit
69 order by `meilleure service après vente` desc
70 limit 1;
71

```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:	Fetch rows:
produit	meilleure service après vente				
Loisirs	8.51				

La typologie de produit qui apporte le meilleur service après vente, est celle des '**Loisirs**' avec une **note moyenne de 8,51**

```

74 -- requete 8: La note moyenne sur l'ensemble des boissons
75
76 • select round(avg(note), 2) as 'note moyenne boisson'
77 from retour_client
78 join produit on retour_client.cle_produit = produit.cle_produit
79 where produit.titre_produit like '%boissons%';
80
81
82

```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
	note moyenne boisson			
►	8.21			

La note moyenne sur l'ensemble des boissons et de **8,21**

```

83  -----
84  -- requete 9: Le classement des jours de la semaine où
85  -- l'expérience client est la meilleure expérience en magasin
86
87
88  • SELECT DAYNAME(date_achat) AS `jour meilleur experience en magasin`, round(avg(note), 2) as note
89  FROM retour_client
90  where libelle_categorie = 'expérience en magasin'
91  group by DAYNAME(date_achat)
92  order by note DESC;
93

```

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

	jour meilleur experience en magasin	note
▶	Saturday	8.34
	Sunday	8.18
	Friday	8.07
	Thursday	8.04
	Wednesday	7.99
	Tuesday	7.95
	Monday	7.74

Le **samedi** est jour de la semaine où l'expérience client en magasin est la meilleure avec **une note moyenne de 8,34**

```

93  -----
94  -- requete 10: Mois sur lequel il y a le plus de retour sur le service après-vente
95
96
97  • select monthname(date_achat) as 'mois', count(libelle_categorie)
98  as 'retour clients service après-vente'
99  from retour_client
100  where libelle_categorie= 'service après-vente'
101  group by monthname(date_achat)
102  order by count(libelle_categorie) desc
103  limit 1;
104

```

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

	mois	retour clients service après-vente
▶	October	55

Le mois d'**octobre** est le mois où il y a le plus de retour sur le service après vente avec **55 retours-clients**

```

106 -- -requete 11: Le pourcentage de recommandations client
107 -- (Comptabiliser le nombre de retours client qui ont répondu "Oui"
108 -- divisé par le nombre de retours total)
109
110 • select sum(recommandation)/count(recommandation)* 100 as 'pourcentage recommandation'
111   from retour_client;
112
113
114

```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
pourcentage recommandation			
70.5			

Un pourcentage de recommandations client élevé qui est de **70,5%**

```

124 -- 12 Magasin qui ont une note inferieure a la moyenne
125
126 • select ref_magasin, round(avg(note), 2) as `note moyenne < moyenne`
127   from retour_client
128   -- where retour_client.ref_magasin= ref_magasin
129   group by ref_magasin
130   having round(avg(note), 2) < (select round(avg(note), 2) as `note moyenne`
131     from retour_client)
132   order by `note moyenne < moyenne` desc;
133

```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
ref_magasin	note moyenne < moyenne		
3	8.04		
47	8.03		
7	8.00		
59	8.00		
53	7.97		
38	7.97		
73	7.97		
79	7.95		
45	7.93		
64	7.93		
13	7.92		
6	7.90		

Liste des magasins qui ont une note inférieure a la moyenne

```

142 -- requete 13: Les typologies produits qui ont amélioré
143 -- leur moyenne entre le 1er et le 2ème trimestre 2021
144
145 • WITH
146     trimestre_1 AS (
147     SELECT
148         typologie_produit,
149         ROUND(AVG(note), 2) AS note_moyenne_t1
150     FROM retour_client
151     JOIN produit ON retour_client.cle_produit = produit.cle_produit
152     WHERE extract(year from date_achat) = 2021
153           AND extract(month from date_achat) BETWEEN 1 AND 3
154     GROUP BY typologie_produit
155     ),
156     trimestre_2 AS (
157     SELECT
158         typologie_produit,
159         ROUND(AVG(note), 2) AS note_moyenne_t2
160     FROM retour_client
161     JOIN produit ON retour_client.cle_produit = produit.cle_produit
162     WHERE extract(year from date_achat) = 2021
163           AND extract(month from date_achat) BETWEEN 4 AND 6
164     GROUP BY typologie_produit
165     )
166 SELECT
167     t1.typologie_produit AS 'type produit',
168     t1.note_moyenne_t1 AS 'note moyenne T1',
169     t2.note_moyenne_t2 AS 'note moyenne T2'
170 FROM
171     trimestre_1 t1
172 JOIN
173     trimestre_2 t2 ON t1.typologie_produit = t2.typologie_produit
174 WHERE
175     t2.note_moyenne_t2 > t1.note_moyenne_t1
176 ORDER BY t1.typologie_produit;
177

```

Result Grid			
	Filter Rows:	Export:	Wrap Cell Content:
	type produit	note moyenne T1	note moyenne T2
▶	Alimentaire	7.99	8.06
	Loisirs	8.00	8.34

Les typologies de produits **Alimentaires et Loisirs** sont celles qui ont amélioré leur moyenne entre le 1<sup>er</sup> et le 2ème trimestre 2021 en passant respectivement **de 7,99 à 8,06 pour l'alimentaire** et de **8,00 à 8,34 pour les loisirs**

```

32  -- - requete 14: CALCUL DU NPS-----
33
34  • select round(promoteur*100/total - detracteur*100/total) as NPS
35  FROM
36  (
37  SELECT
38  sum(case when note > 8
39  then 1
40  else 0 end) as promoteur,
41  sum(case when note <= 6
42  then 1
43  else 0 end) as detracteur,
44  count(note) as total
45  FROM retour_client)as stats;
46
47  -- NPS par source

```

Result Grid	
NPS	
31	

$NPS = (\text{total promoteurs} \times 100 - \text{total detracteurs} \times 100)$

**NPS=31** : Bonne fidélité, mais il y a de la place pour l'amélioration.

```

60  -- requete 15: NPS par source
61
62  • SELECT
63  libelle_source,
64  round((promoteur * 100 / total - detracteur * 100 / total)) AS NPS
65  FROM
66  (SELECT
67  libelle_source,
68  SUM(CASE WHEN note > 8 THEN 1 ELSE 0 END) AS promoteur,
69  SUM(CASE WHEN note <= 6 THEN 1 ELSE 0 END) AS detracteur,
70  COUNT(note) AS total
71  FROM retour_client
72  GROUP BY libelle_source) AS stats;

```

Result Grid	
libelle_source	NPS
▶ téléphone	34
email	30
réseaux sociaux	30

**Indice de satisfaction correcte** sur les différentes sources, email 30%, téléphone 34% et réseau sociaux 30%

```

79  -- 16 : le nombre de retour clients par source
80
81  • select distinct libelle_source as 'source', count(note)as `nombre retour client`
82    from retour_client
83    group by libelle_source
84    order by `nombre retour client`desc;
85

```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
source	nombre retour client			
email	1032			
réseaux sociaux	998			
téléphone	970			

Le nombre de retour client par email est le plus élevé avec **1032 retours**

```

87  -- requete 17: les 5 magasins avec le plus de feedbacks
88
89  • select ref_magasin, count(note)as feedbacks
90    from retour_client
91    group by ref_magasin
92    order by feedbacks desc
93    limit 5;
94
95
96

```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
ref_magasin	feedbacks			
29	55			
6	49			
80	47			
5	45			
83	44			

**Les magasin 29,6,80,5,et 83** sont les magasins qui ont eu le plus de feedbacks



## 5) Cohérence des données



Pour garantir la cohérence des données il est Important de vérifier les contraintes d'intégrité:

Avec la clé primaire, qui permet d'identifier de manière unique les différents enregistrements qui composent une table et qui facilite la mise en relation d'une table avec une autre.

Avec la ou les clés étrangères d'ont le rôle est d'assurer la cohérence des données. Elle permet également de mettre en relation les différentes tables de la BDD. C'est aussi une contrainte qui assure l'intégrité référentielle de celle-ci.

Pour vérifier que les clés étrangères sont bien présentes et que les références sont valides, on peut effectuer une jointure entre les tables concernées.



**Sur MySQL Workbench on peut utiliser des outils pour l'analyse visuelle :**

on peut aussi vérifier les contraintes en accédant **à l'onglet "Inspector"** pour une table donnée, où les informations sur les clés primaires, les clés étrangères et les contraintes de colonnes (comme NOT NULL) sont visibles.

Il faut :

Cliquez sur la table souhaitée dans l'onglet "Navigator" de MySQL Workbench.

Allez à l'onglet "Table".

Sous l'onglet "Columns", on peut voir les informations sur les types de données et les contraintes (NOT NULL, etc.).

Sous l'onglet "Indexes", on peut vérifier les clés primaires et uniques.

Sous l'onglet "Foreign Keys", on peut voir toutes les clés étrangères définies pour la table.

Ce sont les principales vérifications à effectuer pour garantir la qualité des données dans une base de données.

Car, elles permettent de s'assurer que la base de données respecte les contraintes d'intégrité et qu'il n'y a pas de données incohérentes.