



# BestMarket

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

Customer data feedback ou données des retours clients de BestMarket

*Neyna*

*MOHAMED YEHDIH*

*11/07/2024*

# 1) Contexte et expression du besoin



BestMarket est une entreprise de distribution qui envisage d'améliorer la qualité de son réseau de magasins.

Suite à des réclamations et demandes d'informations de plus en plus exigeantes aux prés de leurs clients , BestMarket souhaite analyser les données de retours et avis de clients dans l'objectif d'obtenir des conseils sur comment l'entreprise peut améliorer l'expérience client et mettre en place une stratégie qui permet de fidéliser leurs clients.

## 2) Sauvegarde et stockage de la BDD

### Stratégie de sauvegarde et stockage d'une base de donnée

- ❑ **Choisir une Méthode de Sauvegarde**
  - Copie simple du fichier
  - Utilisation de commandes SQLite
  - Scripts automatisés
- ❑ **Déterminer la Fréquence des Sauvegardes**
  - Sauvegardes quotidiennes
  - Sauvegardes hebdomadaires
- ❑ **Planifier les Sauvegardes**
  - Utiliser un planificateur de tâches
- ❑ **Stocker les Sauvegardes de Manière Sécurisée**
  - Emplacements multiples
  - Chiffrement
- ❑ **Vérifier et Tester les Sauvegardes**
  - Vérification des sauvegardes
  - Tests de restauration

### 3) Méthodologie suivie



- **Etape 1** : Expression des besoins
- **Etape2** : Charger la base de données customer\_data\_feedback dans le SGBDR SQLite studio
- **Etape 3** : Importer de la table ref\_magasin dans la base, mise à jour du schéma de la base de données ainsi que du dictionnaire de donnée en rajoutant les attributs de la table ref\_magasin
- **Etape 4** : Requêtes SQL pour répondre aux besoins clients
- **Etape 5** : La présentation

# Le chargement et les modifications faites sur la base de données

feedback (SQLite 3)

Tableaux (3)

- produit
- ref\_magasin
- retour\_client

Vues

Structure Données Contraintes Index Déclencheurs DDL

feedback Nom de la table : retour\_client

	Nom	Type de données	Clé primaire	Clé étrangère	Unique	Contrôle	Non NULL	Collecter	Généré
1	cle_retour_client	INT	🔑						NULL
2	note	INT							NULL
3	cle_produit	INT		🔗					NULL
4	ref_magasin	INT		🔗					NULL
5	date_achat	DATE							NULL
6	libelle_source	CHAR (50)							NULL
7	libelle_categorie	CHAR (50)							NULL
8	recommandation	CHAR							NULL

feedback (SQLite 3)

Tableaux (3)

- produit
- ref\_magasin
- retour\_client

Vues

Structure Données Contraintes Index Déclencheurs DDL

feedback Nom de la table : produit

	Nom	Type de données	Clé primaire	Clé étrangère	Unique	Contrôle	Non NULL	Collecter	Généré
1	cle_produit	INT	🔑						NULL
2	typologie_produit	CHAR (50)							NULL
3	titre_produit	CHAR (50)							NULL

feedback (SQLite 3)

Tableaux (3)

- produit
- ref\_magasin
- retour\_client

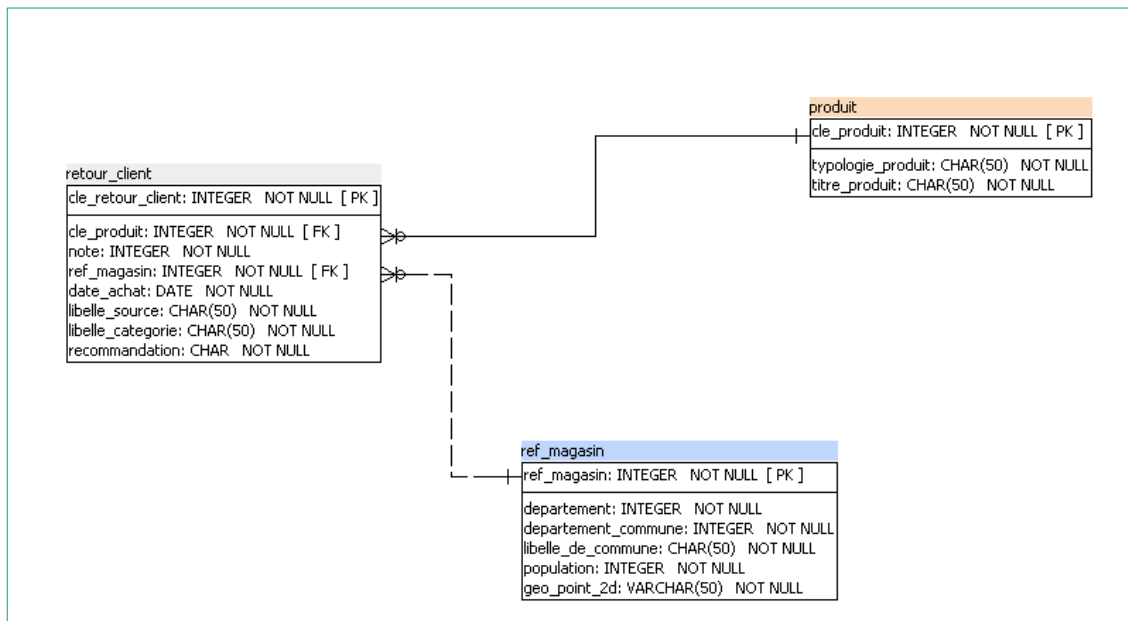
Vues

Structure Données Contraintes Index Déclencheurs DDL

feedback Nom de la table : ref\_magasin

	Nom	Type de données	Clé primaire	Clé étrangère	Unique	Contrôle	Non NULL	Collecter	Généré
1	ref_magasin	INTEGER	🔑						NULL
2	departement	INTEGER							NULL
3	departement_commune	INTEGER							NULL
4	libelle_de_commune	VARCHAR							NULL
5	population	INTEGER							NULL
6	geo_point_2d	VARCHAR							NULL

# Schéma de la base de données



# Dictionnaire de données BestMarket

Tables	Nom du champs	Type de donnée	Taille	Contraint	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 entreprise à votre entourage ?"
	Cle_produit	INT		Clé étrangère	ID des produits
	ref_magasin	INT		Clé étrangère	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)
Table Produit	recommandation	CHAR			Recommandation laissée par le client à la question "Recommandez vous l'entreprise?" True / False
	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	ref_magasin	INT		Clé primaire	ID unique pour les magasins
	departement	INT			Code de departement des magasins
	departement_commune	INT			Code de departement des magasins
	libelle_de_commune	CHAR	50		libelle_de_commune est le nom de la commune des magasins
	population	INT			Population qui visite le magasin
	geo_point_2d	VARCHAR	50		geo_point_2d est la géolocalisation des magasins

## 4) Requêtes SQL et Analyses (1)

**1. Quel est le nombre de retours clients sur la livraison ?**

```
SELECT COUNT(*) AS  
retours_clients_sur_livraison  
FROM retour_client  
WHERE libelle_categorie = 'livraison';
```

retours clients sur livraison
639

**2. Quelle est la liste des notes des clients sur les réseaux sociaux sur les TV**

```
SELECT rc.note AS  
Liste_notes_sur_reseaux_sociaux_pour_TV  
FROM retour_client rc  
JOIN produit p ON rc.cle_produit =  
p.cle_produit  
WHERE rc.libelle_source = 'réseaux sociaux'  
AND p.titre_produit = 'TV';
```

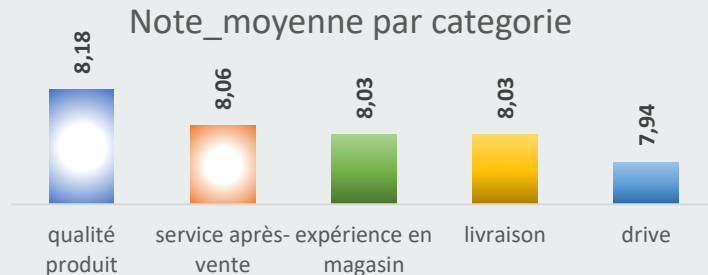
Liste notes sur reseaux sociaux sur les TV	
1	8
2	9
3	10
4	10



## 4) Requêtes SQL et Analyses (2)

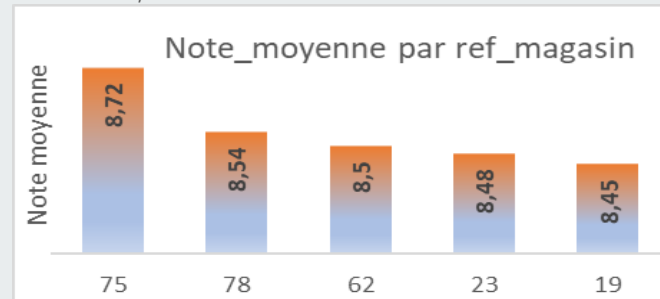
**3. Quelle est la note moyenne pour chaque catégorie de produit ? (Classé de la meilleure à la moins bonne)?**

```
SELECT libelle_categorie, AVG(note) AS  
note_moyenne  
FROM retour_client  
GROUP BY libelle_categorie  
ORDER BY note_moyenne DESC;
```



**4. Quels sont les 5 magasins avec les meilleures notes moyennes ?**

```
SELECT ref_magasin, AVG(note) AS  
note_moyenne  
FROM retour_client  
GROUP BY ref_magasin  
ORDER BY note_moyenne DESC  
LIMIT 5 ;
```



## 4) Requêtes SQL et Analyses (3)

5. Quels sont les magasins qui ont plus de 12 feedbacks sur le drive ?

```
SELECT ref_magasin, COUNT(note) AS  
nombre_feedbacks  
FROM retour_client  
WHERE libelle_categorie = 'drive'  
GROUP BY ref_magasin  
HAVING nombre_feedbacks > 12;
```

ref_magasin	Nb de feedbacks
45	13
63	13
67	14

6. Quel est le classement des départements par note ?

```
SELECT r.departement, ROUND( AVG(rc.note), 2) AS  
note_moyenne  
FROM ref_magasin r  
JOIN retour_client rc ON r.ref_magasin =  
rc.ref_magasin  
GROUP BY r.departement  
ORDER BY note_moyenne DESC;
```

Département	Note moyenne
95	8,14
75	8,11
94	8,06
91	8,05
77	8,04
92	8,03
78	8,02
93	7,94

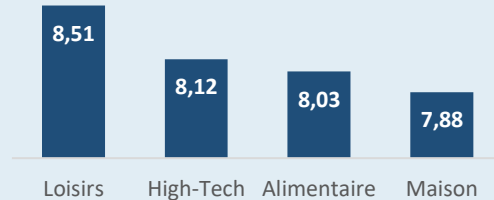
note moyenne  
8.05

## 4) Requêtes SQL et Analyses (4)

**7. Quelle est la typologie de produit qui apporte le meilleur service après-vente ?**

```
SELECT p.typologie_produit, avg(rc.note) AS  
moyenne  
FROM produit p  
JOIN retour_client rc ON p.cle_produit =  
rc.cle_produit  
WHERE rc.libelle_categorie = 'service après-vente'  
GROUP BY p.typologie_produit  
ORDER BY moyenne DESC  
LIMIT 1;
```

typologie produit	moyenne
Loisirs	8.51351351351351



**8. Quelle est la note moyenne sur l'ensemble des boissons ?**

```
SELECT AVG(rc.note) AS note_moyenne  
FROM retour_client rc  
JOIN produit p ON p.cle_produit =  
rc.cle_produit  
WHERE p.titre_produit = 'Boissons';
```

note moyenne 8.07142857142857
----------------------------------

## 4) Requêtes SQL et Analyses (5)

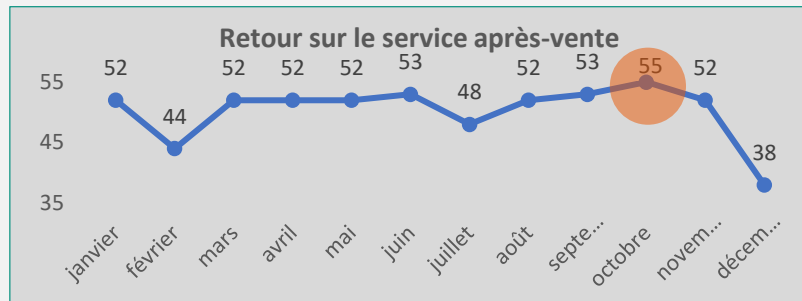
9. le classement des jours de la semaine où l'expérience client est la meilleure expérience en magasin ?

```
SELECT strftime('%w',date_achat) AS jour, AVG(note)
AS moyenne
FROM retour_client
WHERE libelle_categorie = 'expérience en magasin'
GROUP BY jour
ORDER BY moyenne DESC;
```



10. Sur quel mois a-t-on le plus de retour sur le service après-vente ?


```
SELECT strftime('%m',date_achat) AS mois
FROM retour_client
WHERE libelle_categorie = 'service après-vente'
GROUP BY mois
ORDER BY COUNT(*) DESC
LIMIT 1;
```



## 4) Requêtes SQL et Analyses (6)

**11. Quel est le pourcentage de recommandations client ? (Comptabiliser le nombre de retours client qui ont répondu "Oui" divisé par le nombre de retours total)**

```
SELECT COUNT(*)*100 / (SELECT COUNT (*)  
FROM retour_client WHERE recommandation IN  
(0,1)) AS pourcentage_recommandation FROM  
retour_client  
WHERE recommandation = 1 ;
```

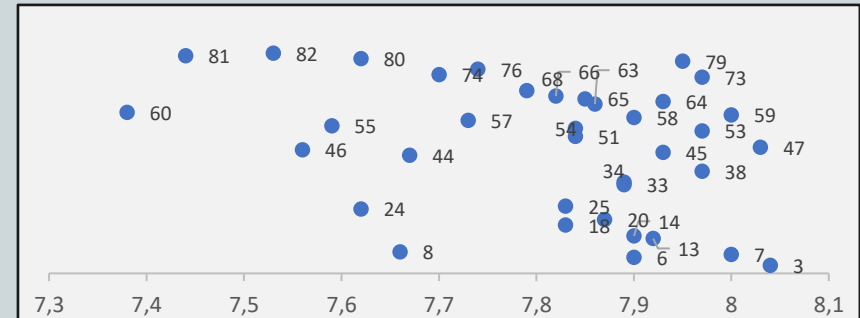
 pourcentage\_recommandation  
90

**12. Quels sont les magasins qui ont une note inférieure à la moyenne?**

```
SELECT ref_magasin, ROUND(AVG(note), 2) AS note,  
(SELECT ROUND(AVG(note), 2)  
FROM retour_client) AS note_moyenne  
FROM retour_client  
GROUP BY ref_magasin  
HAVING note < note_moyenne  
ORDER BY note;
```

note moyenne  
8.05

**Les magasins qui ont une note inférieure à la moyenne**



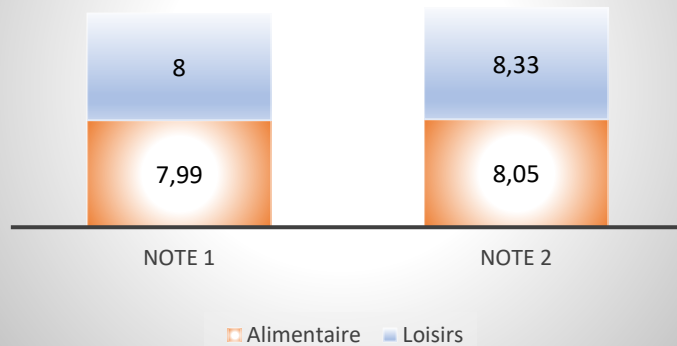
## 4) Requêtes SQL et Analyses (7)

### 13. Quelles sont les typologies produits qui ont amélioré leur moyenne entre le 1er et le 2ème trimestre 2021 ?

```
WITH t1 AS (SELECT p.typologie_produit, AVG(rc.note) AS note1
FROM retour_client rc
JOIN produit p ON rc.cle_produit = p.cle_produit
WHERE rc.date_achat BETWEEN '2021-01-01' AND '2021-03-31'
GROUP BY p.typologie_produit)
```

```
t2 AS (SELECT p.typologie_produit, AVG(rc.note) AS note2
FROM retour_client rc
JOIN produit p ON rc.cle_produit = p.cle_produit
WHERE rc.date_achat BETWEEN '2021-04-01' AND '2021-06-30'
GROUP BY p.typologie_produit)
SELECT typologie_produit, note1, note2 FROM t1
JOIN t2 ON t1.typologie_produit = t2.typologie_produit
WHERE note2 > note1;
```

Typologies produits qui ont amélioré leur moyenne entre le 1er et le 2ème trimestre 2021



## 4) Requêtes SQL et Analyses (8)

### 14.NPS

```
SELECT (promoteurs*1.0/total*100)-(detracteurs*1.0/total*100) AS NPS
FROM (SELECT
(SELECT COUNT(*) FROM retour_client WHERE note BETWEEN 0 AND 6)
AS detracteurs,
(SELECT COUNT (*) FROM retour_client WHERE note BETWEEN 9 AND
10) AS promoteurs,
(SELECT COUNT(*) FROM retour_client) AS total) ;
```

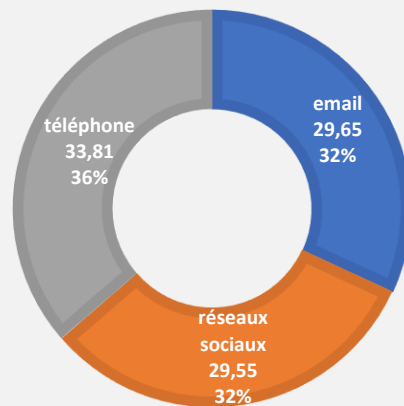
NPS

30.966666666666667

### 15.NPS par source

```
SELECT libelle_source, (promoteurs*1.0/total*100)-
(detracteurs*1.0/total*100) AS NPS
FROM (
SELECT libelle_source,
count(case when note >=9 then 1 else null end) AS promoteurs,
count(case when note <=6 then 1 else null end) AS detracteurs,
count(*) AS total
FROM retour_client
GROUP BY libelle_source );
```

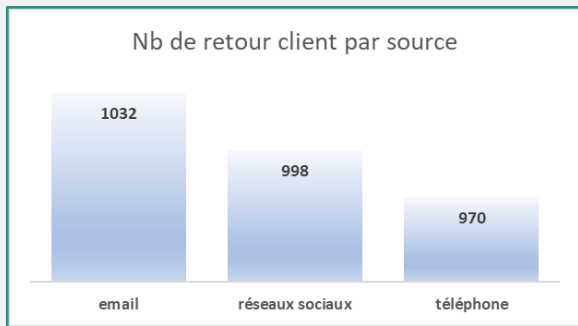
NPS PAR SOURCE



## 4) Requêtes SQL et Analyses (9)

### 16. Quel est le nombre de retour clients par source ?

```
SELECT libelle_source, COUNT(*) AS  
nb_retour_client  
FROM retour_client  
GROUP BY libelle_source ;
```



### 17. Exemple 2 : Quels sont les 5 magasins avec le plus de feedbacks ?

```
SELECT ref_magasin, COUNT(note) AS  
nb_feedbacks  
FROM retour_client  
GROUP BY ref_magasin  
ORDER BY nb_feedbacks DESC  
LIMIT 5;
```

ref_magasin	nb_feedbacks
29	55
6	49
80	47
5	45
83	44



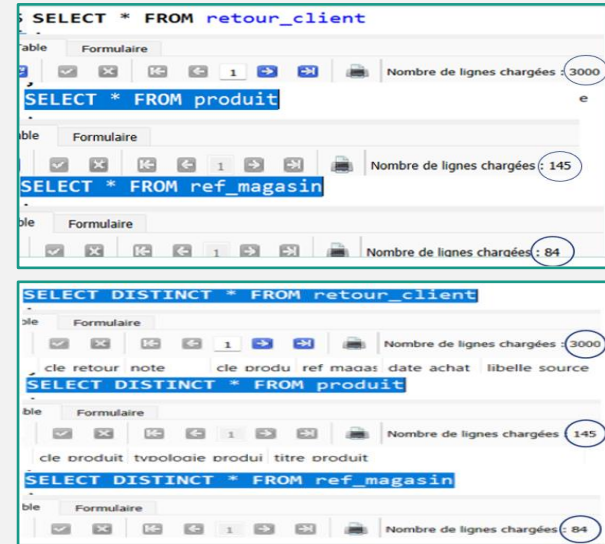
## 5) Cohérence des données

Les données sont globalement cohérentes,

Les clés étrangères avaient bien une correspondance dans d'autres tables.

Pas de doublons, ils sont vérifiés avec la commande DISTINCT.

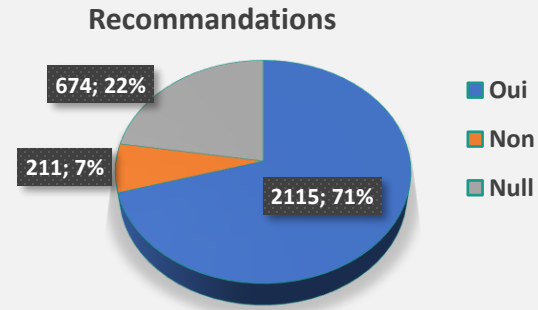
Les requêtes SQL ont été vérifiées, elles sont fonctionnelles et ne retournent pas des erreurs.



## Axes d'analyses pertinentes (2)

Compte tenu du fait que 22% des clients n'ont pas donné leur avis cela a impacter une partie des analyses de données.

Afin que les analyses sur les recommandations soient pertinentes, il faudrait analyser les recommandations par catégorie et par magasin et inciter les clients à donner leurs avis lors des prochaines enquêtes de satisfactions



# Axes d'analyses pertinentes (1)

Calculer les NPS (net promoter score) par catégorie pour mieux suivre la satisfaction client et prendre des décisions qui peuvent améliorer l'expérience clients.

