



BestMarket

Manipulation d'une base de données SQL pour suivre satisfaction client

Samuel

OSENAT

3 janvier 2025

1) Contexte et expression du besoin



- Projet RetailInsight360° enjeu stratégique porté par la direction
- Besoin de mieux comprendre la clientèle pour améliorer qualité de service des magasins
- Mise à disposition de fichiers de données, à partir desquels on crée des indicateurs qui permettront de répondre
- Les indicateurs utilisés sont la note moyenne accordée par les clients, le NPS, le nombre de retours.

2) Sauvegarde et stockage de la BDD



- Collecter, exploiter et sauvegarder les données en respectant les principes du RGPD
- Définir une durée de conservation des données
- Porter attention à la volumétrie des données stockées et mettre en place des mesures correctives si elles sont trop volumineuses (sans utilité particulière)
- Limiter l'accès aux données dans l'entreprise pour ne pas qu'elles se diffusent sans raison valable

3) Méthodologie suivie

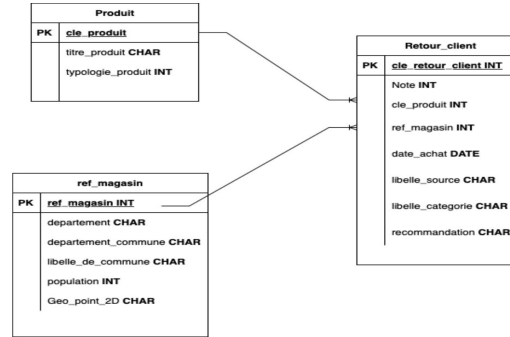
- Utilisation du logiciel Dbeaver qui sert d'interface avec le moteur de données SQLite
- Extraction des tables retour_client et produit à partir d'un script SQL fourni
- Ajout de la table magasin à partir d'un fichier CSV

Dictionnaire de données suite à l'ajout de la nouvelle table :

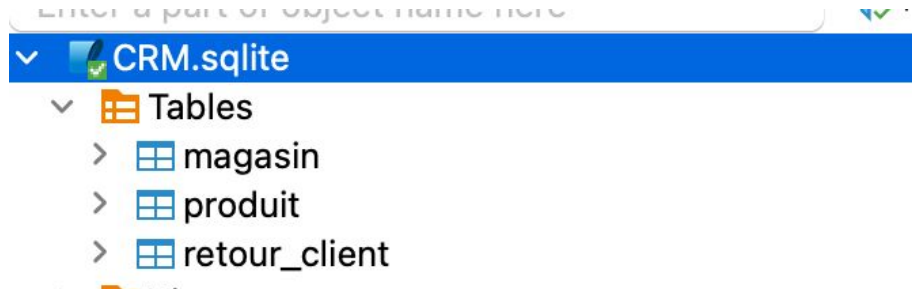
	Nom du champs	Type de données	Taille	Contrainte	Description
Table Retour client	cle_retour_client	INT		Clé primaire	ID unique pour les retours clients
	note	INT			Note donnée par le client, comprise entre 0 et 10, la note est la réponse à la question : "Sur une échelle de 0 à 10 quelle est la probabilité que vous recommandiez notre entreprise à votre entourage ?"
	Clé_produit	INT			ID des produits
	ref_magasin	INT			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	ref_magasin	INT		Clé primaire	ID unique pour les magasins
	departement	CHAR			département dans lequel se situe le magasin
	departement_commune	CHAR			Code postal
	libelle de commune	CHAR			Nom de la commune
	population	INT			Nombre d'habitants de la commune
	Geo_point_2D	CHAR			Coordonnées géographiques

3) Méthodologie suivie

Schema relationnel :



Base de données dans Dbeaver:



4) Requêtes SQL et Analyses

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

```
SELECT  
COUNT(cle_retour_client) as 'Nombre retour client'  
FROM retour_client  
WHERE libelle_categorie = 'livraison' ;
```

SELECT COUNT(cle_retour_client) | Ent

	123 COUNT(cle_retour_client)
1	639

2) Quelle est la liste des notes des clients sur les réseaux sociaux sur les TV ?

```
SELECT  
cle_retour_client,  
note  
FROM retour_client rc  
JOIN produit p ON rc.cle_produit = p.cle_produit  
WHERE libelle_source = 'réseaux sociaux' AND titre_produit = 'TV'  
GROUP BY note;
```

retour_client 1 ×

SELECT cle_retour_client, note FRC | Entrez une e

	123 cle_retour_client	123 note
1	158 040	8
2	157 495	9
3	156 795	10

4) Requêtes SQL et Analyses

3) Quelle est la note moyenne pour chaque catégorie de produit (classée de la meilleure à la moins bonne)

```
SELECT  
typologie_produit,  
ROUND(AVG(note),2) as note_moyenne  
FROM retour_client rc  
JOIN produit p ON rc.cle_produit = p.cle_produit  
GROUP BY typologie_produit  
ORDER BY AVG(note) DESC
```

produit 1 X		
SELECT typologie_produit, ROUND(AVG(n		
	A-Z typologie_produit	123 note_moyenne
1	High-Tech	8,16
2	Loisirs	8,09
3	Alimentaire	8,04
4	Maison	7,85

4) Quels sont les 5 magasins avec les meilleures notes moyennes

```
SELECT  
m.ref_magasin,  
ROUND(AVG(note),2) as note_moyenne  
FROM retour_client rc  
JOIN magasin m  
ON rc.ref_magasin = m.ref_magasin  
GROUP BY m.ref_magasin  
ORDER BY AVG(note) DESC  
LIMIT 5
```

magasin 1 X		
SELECT m.ref_magasin, ROUND(AVG(r		
	123 ref_magasin	123 note_moyenne
1	75	8,73
2	78	8,55
3	62	8,5
4	23	8,48
5	19	8,45

4) Requêtes SQL et Analyses

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

```
SELECT ,m.ref_magasin,  
COUNT(rc.cle_retour_client) as nombre_feedback  
FROM retour_client rc  
JOIN magasin m  
ON rc.ref_magasin = m.ref_magasin  
WHERE libelle_categorie = 'drive'  
GROUP BY m.ref_magasin  
HAVING COUNT(rc.cle_retour_client) > 12
```

SQL: `SELECT m.ref_magasin, COUNT(rc.cle_retour_client) as nombre_feedback`

	123 ref_magasin	123 nombre_feedback
1	45	13
2	63	13
3	67	14

6) Quel est le classement des départements par note ?

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

SQL: `SELECT m.departement, ROUND(AVG(rc.note), 2) as note_moyenne`

	123 departement	123 note_moyenne
1	95	8,14
2	75	8,11
3	94	8,06
4	91	8,05
5	77	8,04
6	92	8,03
7	78	8,02
8	93	7,94

4) Requêtes SQL et Analyses

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

SELECT

typologie_produit,
ROUND(AVG(note),2) as note_moyenne

FROM retour_client rc

JOIN produit p **ON** rc.cle_produit = p.cle_produit

WHERE libelle_categorie = 'service après-vente'

GROUP BY typologie_produit

ORDER BY AVG(note) **DESC**

LIMIT 1;

SELECT typologie_produit, ROUND(AVG(note),2) as note_moyenne

Tableau	A-Z typologie_produit	123 note_moyenne
1	Loisirs	8,51

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

SELECT

ROUND(AVG(note),2) as note_moyenne

FROM retour_client rc

JOIN produit p **ON** rc.cle_produit = p.cle_produit

WHERE titre_produit **LIKE** 'Boissons%';

Résultats 1 X

SELECT ROUND(AVG(note),2) as note_moyenne

123 note_moyenne
8,32

4) Requêtes SQL et Analyses

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

```
SELECT  
STRFTIME('%w', date_achat) as jour_semaine,  
ROUND(AVG(note),2) as note_moyenne  
FROM retour_client  
WHERE libelle_categorie = 'expérience en magasin'  
GROUP BY jour_semaine  
ORDER BY note_moyenne DESC;
```

T **SELECT STRFTIME('%w', date_achat) :** | ⌵ ⌶ ⌴ Entrez une ex

	A-Z jour_semaine	123 note_moyenne
1	6	8,34
2	0	8,18
3	5	8,07
4	4	8,04
5	3	7,99
6	2	7,95
7	1	7,74

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

```
SELECT  
STRFTIME('%m', date_achat) as mois,  
COUNT(cle_retour_client) as nombre_retour_client  
FROM retour_client  
WHERE libelle_categorie = 'service après-vente'  
GROUP BY mois  
ORDER BY nombre_retour_client DESC  
LIMIT 1;
```

<T **SELECT STRFTIME('%m', date_ach:** | ⌵ ⌶ ⌴ Entrez une exp

	A-Z mois	123 n
1	10	55

Ctrl+click pour ouvrir la cons

4) Requêtes SQL et Analyses

11) Quel est le pourcentage de recommandations clients ?

```
SELECT  
ROUND(100.0 * SUM(CASE WHEN recommandation = '1' THEN 1 ELSE 0 END) / COUNT(recommandation),2) AS  
pourcentage_recommandation  
FROM retour_client  
WHERE recommandation IN ('0', '1');
```

	123 pourcentage_recommandation
1	90,93

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

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

retour_client 1 X		
SELECT ref_magasin, ROUND(AVG(not) Entrez une e		
Tableau	123 ref_magasin	123 note_moyenne
1	3	8,04
2	47	8,03
3	59	8
4	7	8
5	73	7,97
6	53	7,97
7	38	7,97
8	79	7,95
9	64	7,93

4) Requêtes SQL et Analyses

13) Quelles sont les typologies produits qui ont amélioré leur moyenne entre le 1er et le 2e trimestre 2021

SELECT

```
p.typologie_produit,  
ROUND(AVG(CASE WHEN strftime('%Y-%m', rc.date_achat) BETWEEN '2021-01' AND '2021-03' THEN rc.note  
ELSE NULL END),2) AS moyenne_T1,  
ROUND(AVG(CASE WHEN strftime('%Y-%m', rc.date_achat) BETWEEN '2021-04' AND '2021-06' THEN rc.note  
ELSE NULL END),2) AS moyenne_T2,
```

```
ROUND(  
  ((AVG(CASE WHEN strftime('%Y-%m', rc.date_achat) BETWEEN '2021-04' AND '2021-06' THEN rc.note  
ELSE NULL END)  
  - AVG(CASE WHEN strftime('%Y-%m', rc.date_achat) BETWEEN '2021-01' AND '2021-03' THEN rc.note ELSE  
NULL END))  
  / AVG(CASE WHEN strftime('%Y-%m', rc.date_achat) BETWEEN '2021-01' AND '2021-03' THEN rc.note ELSE  
NULL END)) * 100, 2) AS 'evolution%'  
FROM retour_client rc  
JOIN produit p ON rc.cle_produit = p.cle_produit  
GROUP BY p.typologie_produit  
HAVING moyenne_T2 > moyenne_T1;
```

Tableau	Ctrl+click pour ouvrir la console SQL			
		moyenne_T1	moyenne_T2	evolution%
1	Alimentaire	7,99	8,06	0,81
2	Loisirs	8	8,34	4,24

4) Requêtes SQL et Analyses

14) NPS

WITH Calcul AS (

SELECT

ROUND(100.0 * (SUM(CASE WHEN note >= 9 THEN 1 ELSE 0 END) * 1.0 / COUNT(note)), 1) as "%Promoteurs",

ROUND(100.0 * (SUM(CASE WHEN note <= 6 THEN 1 ELSE 0 END) * 1.0 / COUNT(note)), 1) as "%Détracteurs"

FROM retour_client rc

)

SELECT

"%Promoteurs",

"%Détracteurs",

("%Promoteurs" - "%Détracteurs") AS NPS

FROM Calcul;

Résultats 1 ×

WITH Calcul AS (SELECT ROUND(100.0, 1) AS "%Promoteurs", ROUND(100.0, 1) AS "%Détracteurs", (ROUND(100.0, 1) - ROUND(100.0, 1)) AS NPS FROM Calcul)

	123 %Promoteurs	123 %Détracteurs	123 NPS
1	40	9	31

4) Requêtes SQL et Analyses

15) NPS par source

WITH *Calcul* **AS** (

SELECT

libelle_source,

ROUND(100.0 * (SUM(CASE WHEN note >= 9 THEN 1 ELSE 0 END) * 1.0 / COUNT(note)), 1) **as** "%Promoteurs",

ROUND(100.0 * (SUM(CASE WHEN note <= 6 THEN 1 ELSE 0 END) * 1.0 / COUNT(note)), 1) **as** "%Détracteurs"

FROM retour_client rc

GROUP BY libelle_source

)

SELECT

libelle_source **AS** "Source",

"%Promoteurs",

"%Détracteurs",

("%Promoteurs" - "%Détracteurs") **AS** NPS

FROM *Calcul*;

ITH Calcul AS (SELECT libelle_sourc | Entrez une expression SQL pour filtrer les résu

	A-Z source	123 %Promoteurs	123 %Détracteurs	123 NPS
	email	37,5	7,8	29,7
	réseaux sociaux	41,1	11,5	29,6
	téléphone	41,5	7,7	33,8

4) Requêtes SQL et Analyses

16) NPS par département

```
WITH Calcul_magasin AS (  
SELECT  
ROUND(100.0 *(SUM(CASE WHEN note >= 9 THEN 1 ELSE 0 END) * 1.0/COUNT(note)),1) as "%Promoteurs",  
ROUND(100.0 *(SUM(CASE WHEN note <= 6 THEN 1 ELSE 0 END) * 1.0/COUNT(note)),1) as "%Détracteurs",  
ref_magasin  
FROM retour_client rc  
GROUP BY ref_magasin  
)
```

```
Calcul_NPS AS (  
SELECT  
ref_magasin,  
"%Promoteurs",  
"%Détracteurs",  
("%Promoteurs" - "%Détracteurs") AS NPS  
FROM Calcul_magasin)
```

```
SELECT  
m.departement,  
COUNT(m.ref_magasin) AS "Nombre de magasin",  
ROUND(AVG(NPS),2) as NPS_MOYEN  
FROM Calcul_NPS cn  
JOIN magasin m ON cn.ref_magasin = m.ref_magasin  
GROUP BY m.departement  
ORDER BY AVG(NPS) DESC;
```

magasin 1 X

WITH Calcul_magasin AS (SELECT RO | Entrez une expression SQL pour filtrer les

	123 departement	123 Nombre de magasin	123 Nps_moyen
1	94	5	33,54
2	75	13	33,18
3	92	9	32,77
4	91	12	32,6
5	95	11	31,61
6	77	12	29,91
7	78	14	28,89
8	93	7	26,13

4) Requêtes SQL et Analyses

17) Pire jour de l'année en terme de note

```
SELECT  
    date_achat,  
    STRFTIME('%w', date_achat),  
    AVG(note) AS note_moyenne  
FROM retour_client  
GROUP BY date_achat  
ORDER BY note_moyenne ASC  
LIMIT 1;
```

retour_client 1 X

SELECT date_achat, STRFTIME('%w', c) | Entrez une expression SQL pour filtrer les

	A-Z date_achat	A-Z jour_de_la_semaine	123 note_moyenne
1	2021-11-13	6	5,44

5) Cohérence des données



- Données cohérentes dans l'ensemble
- Exceptée colonne "recommandation" de la table retour_client où l'on retrouve des valeurs nulles