



BestMarket

Titre du projet : Customer Data feedback

Prénom : Benoit

Nom : Ah-see

Date : 10/04/2024

1) Contexte et expression du besoin

- ✓ **Projet stratégique de la direction : RetailInsight360**
- ✓ **Projet “Customer Data Feedback” du service-client**
- ✓ **Objectif : conseiller la direction pour bâtir la meilleure expérience d’achat par le client afin de le faire revenir**
- ✓ **Mission : analyser une BDD pour répondre à une liste de questions du service-client**
- ✓ **Suivre l’indicateur NPS et proposer des indicateurs supplémentaires pour améliorer l’expérience client**
- ✓ **Rappeler les bonnes pratiques en terme de sauvegarde, stockage et accès aux données de la base de données**

2) Sauvegarde et stockage de la BDD



- Faire de la sécurité un enjeu partagé et porté par l'équipe dirigeante
- Sécuriser le matériel (serveurs, matériel de sauvegarde...) dans des endroits fermés avec des accès limités et des alarmes anti-intrusion
- Effectuer des sauvegardes fréquentes des données
- Stocker au moins une sauvegarde sur un site géographiquement distinct du site d'exploitation dans des coffres ignifugés et étanches
- Isoler au moins une sauvegarde hors ligne, déconnectée du réseau de l'entreprise
- Tester régulièrement l'intégrité des sauvegardes et la capacité de les restaurer

2) Sauvegarde et stockage de la BDD

- Prévoir un plan de continuité et de reprise d'activité
- Prévoir les procédures et les responsabilités internes pour la gestion des incidents, dont la procédure de notification aux régulateurs des violations de données personnelles (RGPD)
- Appliquer la règle dite « 3 – 2 – 1 », état de l'art en matière de sauvegarde, qui consiste à disposer de 3 copies des données, stocker sur 2 supports différents, dont 1 hors ligne
- Adopter une politique de mot de passe rigoureuse
- Concevoir une procédure de création et de suppression des comptes utilisateurs
- Sécuriser les postes de travail
- Sensibiliser les utilisateurs aux « risques informatiques »

3) Méthodologie suivie

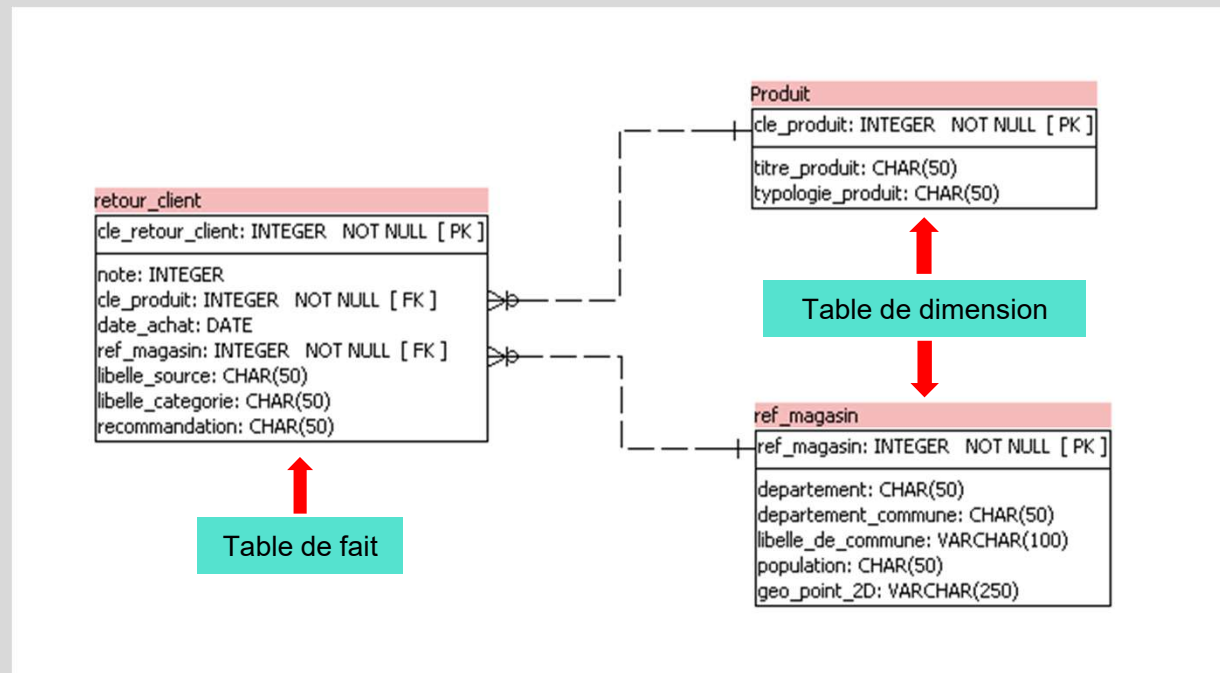
Dictionnaire des données

	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 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)
	recommandation	CHAR	50		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	CHAR	50		Typologie des produits (Alimentaire, High-tech etc...)
Table Ref_magasin	ref_magasin	INT		Clé primaire	ID des magasins
	departement	CHAR	50		Numéro du département du magasin
	departement_commune	CHAR	50		Concatenation du code département et du code commune
	libelle_de_commune	VARCHAR	100		Nom de la commune du magasin
	population	CHAR	50		Nombre d'habitants dans la commune
	geo_point_2d	VARCHAR	250		Position géographique du magasin par rapport à la latitude et à la longitude



3) Méthodologie suivie

Schéma relationnel en 3NF avec le logiciel SQL Power Architect



3) Méthodologie suivie

Vérification de la base de données extraite

Query 1 x

Limit to 1000 rows

```
1 • select * from retour_client;
```

Result Grid

	de_retour_client	note	de_produit	ref_magasin	date_achat	libelle_source	libelle_categorie	recommandation
▶	156732	8	47	47	2021-08-14	téléphone	service après-vente	1
	156733	7	133	6	2021-05-12	téléphone	livraison	1
	156734	10	143	69	2021-06-04	email	expérience en magasin	1
	156735	10	104	47	2021-12-26	réseaux sociaux	drive	
	156736	8	28	54	2021-07-25	email	livraison	1
	156737	7	89	71	2021-12-17	réseaux sociaux	drive	

Query 1 x

Limit to 1000 rows

```
1 • select * from produit;
```

Result Grid

	de_produit	typologie_produit	titre_produit
▶	1	Alimentaire	Additifs alimentaires
	2	Alimentaire	Aliments à base de fruits et de légumes
	3	Alimentaire	Aliments à base de plantes en conserve
	4	Alimentaire	Aliments à base de plantes frais
	5	Alimentaire	Aliments à base de plantes séchées
	6	Alimentaire	Aliments à base de plantes surgelés
	7	Alimentaire	Aliments d'origine végétale
	8	Alimentaire	Aliments et boissons à base de végétaux
	9	Alimentaire	Amuse-gueules
	10	Alimentaire	Bananes
	11	Alimentaire	Beurres de cacahuètes
	12	Alimentaire	Bières

La requête SQL «select * from 'nom du tableau'» permet de vérifier la présence de toutes les colonnes avec les données des 2 tableaux

3) Méthodologie suivie

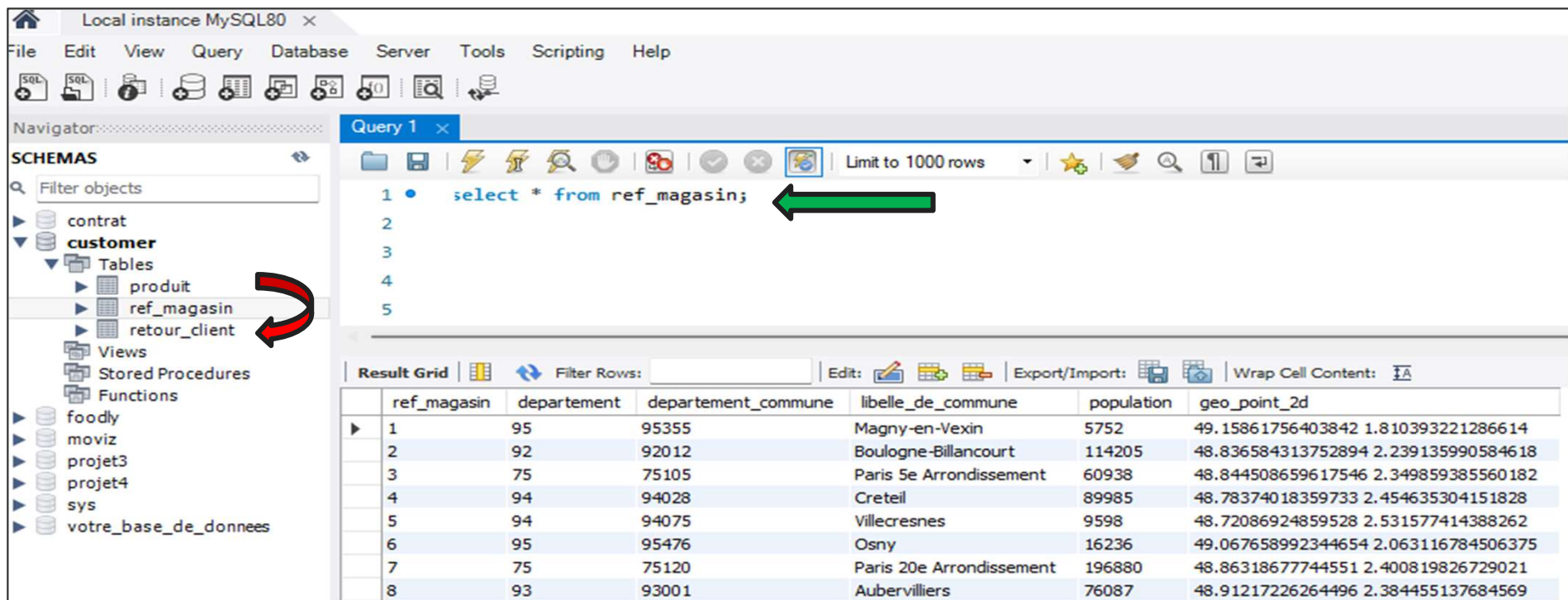
Rajout de la table ref_région dans la BDD

```
1 • CREATE TABLE ref_magasin (  
2     ref_magasin INT PRIMARY KEY,  
3     departement CHAR(50),  
4     departement_commune CHAR(50),  
5     libelle_de_commune VARCHAR(100),  
6     population CHAR(50),  
7     geo_point_2d VARCHAR(250)  
8 );
```

```
1 • LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/ref_magasin.csv'  
2 INTO TABLE ref_magasin  
3 FIELDS TERMINATED BY ';' ENCLOSED BY ''  
4 LINES TERMINATED BY '\n'  
5 IGNORE 1 LINES;
```

La requête SQL permet de rajouter la table ref_magasin dans la base de données et ensuite d'intégrer les données du fichier csv

3) Méthodologie suivie



The screenshot shows the MySQL Workbench interface. On the left, the 'SCHEMAS' pane displays a tree view of databases. The 'customer' database is expanded, showing tables: 'produit', 'ref_magasin', and 'retour_client'. A red arrow points from 'ref_magasin' to the query editor. The query editor, titled 'Query 1', contains the SQL statement: `select * from ref_magasin;`. A green arrow points to this statement. Below the query editor, the 'Result Grid' displays the results of the query. The grid has 7 columns: 'ref_magasin', 'departement', 'departement_commune', 'libelle_de_commune', 'population', and 'geo_point_2d'. It shows 8 rows of data.

	ref_magasin	departement	departement_commune	libelle_de_commune	population	geo_point_2d
1	1	95	95355	Magny-en-Vexin	5752	49.15861756403842 1.810393221286614
2	2	92	92012	Boulogne-Billancourt	114205	48.836584313752894 2.239135990584618
3	3	75	75105	Paris 5e Arrondissement	60938	48.844508659617546 2.349859385560182
4	4	94	94028	Creteil	89985	48.78374018359733 2.454635304151828
5	5	94	94075	Villecresnes	9598	48.72086924859528 2.531577414388262
6	6	95	95476	Osny	16236	49.067658992344654 2.063116784506375
7	7	75	75120	Paris 20e Arrondissement	196880	48.86318677744551 2.400819826729021
8	8	93	93001	Aubervilliers	76087	48.91217226264496 2.384455137684569

La requête SQL «select * from 'ref _ magasin'» permet de vérifier le rajout dans la BDD. Les 3 tables sont bien présentes et les requêtes SQL peuvent démarrer...

4) Requêtes SQL et Analyses

- 1) Nombre de retours clients sur la livraison

```
1 • SELECT COUNT(*) AS cle_retour_client
2   FROM retour_client
3   WHERE libelle_categorie = 'livraison';
```

Result Grid		Filter
	cle_retour_client	
▶	639	

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

```
1 • SELECT note, COUNT(*) AS cle_retour_client
2   FROM retour_client
3   WHERE libelle_source = 'Reseaux sociaux'
4   GROUP BY note
5   order by note DESC;
```

Result Grid			Filter
	note	cle_retour_client	
▶	10	179	
	9	231	
	8	280	
	7	193	
	6	75	
	5	8	
	4	10	
	3	1	
	2	9	
	1	12	

4) Requêtes SQL et Analyses

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

```
1 • SELECT P.typologie_produit AS categorie_produit, ROUND(AVG(RC.note), 2) AS note_moyenne_arrondie
2 FROM Retour_client RC
3 JOIN Produit P ON RC.Cle_produit = P.cle_produit
4 GROUP BY P.typologie_produit
5 ORDER BY note_moyenne_arrondie DESC;
6
```

Result Grid		Filter Rows:
	categorie_produit	note_moyenne_arrondie
▶	High-Tech	8.16
	Loisirs	8.09
	Alimentaire	8.04
	Maison	7.85

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

```
1 • SELECT RM.libelle_de_commune AS magasin, ROUND(AVG(RC.note), 2) AS note_moyenne
2 FROM Retour_client RC
3 JOIN Ref_magasin RM ON RC.ref_magasin = RM.ref_magasin
4 GROUP BY RM.libelle_de_commune
5 ORDER BY note_moyenne DESC
6 LIMIT 5;
7
```

Result Grid		Filter Rows:
	magasin	note_moyenne
▶	Paris 14e Arrondissement	8.73
	Saint-Pierre-du-Perray	8.55
	Paris 19e Arrondissement	8.50
	Paris 11e Arrondissement	8.48
	Coulommiers	8.45

4) Requêtes SQL et Analyses

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

```
1 • SELECT RM.libelle_de_commune AS magasin, COUNT(*) AS cle_retour_client
2 FROM Retour_client RC
3 JOIN Ref_magasin RM ON RC.ref_magasin = RM.ref_magasin
4 WHERE RC.libelle_categorie = 'Drive'
5 GROUP BY RM.libelle_de_commune
6 HAVING COUNT(*) > 12;
7
```

Result Grid		Filter Rows:
magasin	de_retour_client	
Eragny	14	
Ivry-sur-Seine	13	
Paris 12e Arrondissement	13	

- 6) Quel est le classement des départements par note?

```
1 • SELECT RM.departement AS departement, ROUND(AVG(RC.note), 2) AS note_moyenne
2 FROM retour_client RC
3 JOIN Ref_magasin RM ON RC.ref_magasin = RM.ref_magasin
4 GROUP BY RM.departement
5 ORDER BY note_moyenne DESC;
6
```

Result Grid		Filter Rows:
departement	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	

4) Requêtes SQL et Analyses

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

```
1 • SELECT P.typologie_produit AS typologie_produit, ROUND(AVG(RC.note), 2) AS note_moyenne
2 FROM retour_client RC
3 JOIN Produit P ON RC.Cle_produit = P.cle_produit
4 WHERE RC.libelle_categorie = 'service après-vente'
5 GROUP BY P.typologie_produit
6 ORDER BY note_moyenne DESC
7 LIMIT 1;
```

Result Grid		Filter Rows:
typologie_produit	note_moyenne	
Loisirs	8.51	

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

```
1 • SELECT ROUND(AVG(RC.note), 2) AS note_moyenne_boissons
2 FROM retour_client RC
3 JOIN Produit P ON RC.Cle_produit = P.cle_produit
4 WHERE P.titre_produit LIKE '%boissons%' OR P.typologie_produit = 'Sodas';
5
6
```

Result Grid		Filter Rows:
note_moyenne_boissons		
8.21		

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?

```
1 • SELECT DAYNAME(date_achat) AS jour_semaine, ROUND(AVG(RC.note), 2) AS note_moyenne
2 FROM retour_client RC
3 WHERE RC.libelle_categorie = 'expérience en magasin'
4 GROUP BY DAYNAME(date_achat)
5 ORDER BY note_moyenne DESC;
6
```

Result Grid		Filter Rows:
	jour_semaine	note_moyenne
▶	Saturday	8.34
	Sunday	8.18
	Friday	8.07
	Thursday	8.04
	Wednesday	7.99
	Tuesday	7.95
	Monday	7.74

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

```
1 • SELECT MONTH(date_achat) AS mois, COUNT(*) AS cle_retour_client
2 FROM retour_client
3 WHERE libelle_categorie = 'service après-vente'
4 GROUP BY mois
5 ORDER BY cle_retour_client DESC
6 LIMIT 1;
```

Result Grid		Filter Rows:
	mois	cle_retour_client
▶	10	55

4) Requêtes SQL et Analyses

- 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)

```
1 • SELECT
2     ROUND((SUM(recommandation = 1) / COUNT(*) * 100), 2) AS pourcentage_recommandations_oui
3 FROM
4     retour_client;
```

Result Grid		Filter Rows:
	pourcentage_recommandations_oui	
▶	70.50	

- 12) Quelles sont les typologies-produits qui ont amélioré leur moyenne entre le 1er et le 2ème trimestre 2021 ?

```
1 • SELECT
2     typologie_produit,
3     AVG(CASE WHEN MONTH(date_achat) BETWEEN 1 AND 3 THEN note ELSE NULL END) AS moyenne_trimestre1,
4     AVG(CASE WHEN MONTH(date_achat) BETWEEN 4 AND 6 THEN note ELSE NULL END) AS moyenne_trimestre2
5 FROM
6     retour_client RC
7 JOIN
8     Produit P ON RC.cle_produit = P.cle_produit
9 WHERE
10    YEAR(date_achat) = 2021
11 GROUP BY
12    typologie_produit
13 HAVING
14    moyenne_trimestre2 > moyenne_trimestre1;
```

Result Grid		Filter Rows:	Export:	Wrap
	typologie_produit	moyenne_trimestre1	moyenne_trimestre2	
▶	Alimentaire	7.9932	8.0582	
	Loisirs	8.0000	8.3393	

4) Requêtes SQL et Analyses

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

```
1 • SELECT
2     RM.libelle_de_commune
3 FROM
4     (
5     SELECT
6         ref_magasin,
7         AVG(note) AS moyenne_magasin
8     FROM
9         retour_client
10    GROUP BY
11        ref_magasin
12    ) AS RC
13 JOIN
14     Ref_magasin RM ON RC.ref_magasin = RM.ref_magasin
15 WHERE
16     RC.moyenne_magasin < (SELECT AVG(note) FROM retour_client);
17
```

Freneuse
Osny
Paris 2e Arrondissement
Paris 5e Arrondissement
Saint-Germain-les-Arpajon
Carrieres-sur-Seine
Paris 17e Arrondissement
Paris 15e Arrondissement
Aubervilliers
Ozoir-la-Ferriere
Paris 20e Arrondissement
Ivry-sur-Seine
Puteaux
Versailles

Longpont-sur-Orge
Conflans-Sainte-Honorine
Paris 1er Arrondissement
Lognes
Levallois-Perret
Paris 18e Arrondissement
Herblay
Rosny-sous-Bois
Coignieres
Chelles
Montgeron
Buchelay
Nanterre

4) Requêtes SQL et Analyses

FOCUS SUR NPS – NET PROMOTER SCORE

• 14) NPS

```
SELECT
  ROUND(
    (SUM(CASE WHEN note BETWEEN 9 AND 10 THEN 1 ELSE 0 END) / COUNT(*) * 100) -
    (SUM(CASE WHEN note BETWEEN 0 AND 6 THEN 1 ELSE 0 END) / COUNT(*) * 100)
    , 1) AS NPS
FROM
  retour_client;
```

Result Grid	
	NPS
▶	31.0

- ✓ *NPS = pourcentage de réponse avec une note entre 9 et 10 - pourcentage de réponse entre 0 et 6*
- ✓ *Nombre de clients entre 0 et 6 = 271 sur 3 000, soit 9 %*
- ✓ *Nombre de clients entre 9 et 10 = 1200 sur 3000 soit 40 %*
- ✓ *NPS = 40 – 9 = 31*

4) Requêtes SQL et Analyses

- 15) NPS par source

```
1 • SELECT
2     libelle_source,
3     ROUND(
4         (SUM(CASE WHEN note BETWEEN 9 AND 10 THEN 1 ELSE 0 END) / COUNT(*) * 100) -
5         (SUM(CASE WHEN note BETWEEN 0 AND 6 THEN 1 ELSE 0 END) / COUNT(*) * 100)
6     , 1) AS NPS
7 FROM
8     retour_client
9 WHERE
10    libelle_source IN ('téléphone', 'email', 'réseaux sociaux')
11 GROUP BY
12    libelle_source;
```

Result Grid		Filter Ro
	libelle_source	NPS
▶	téléphone	33.8
	email	29.7
	réseaux sociaux	29.6

Le NPS pour chaque source, indicateur clé pour Olivier, responsable du service client, pourra être transmis à la fréquence souhaitée en un clic, si les données sont transmises !!!

4) Requêtes SQL et Analyses

- 16) Quel est le nombre de retour clients par source ?

```
1 • SELECT
2     libelle_source,
3     COUNT(*) AS nombre_retours
4 FROM
5     retour_client
6 GROUP BY
7     libelle_source;
8
```

Result Grid		Filter Rows:
	libelle_source	nombre_retours
▶	téléphone	970
	email	1032
	réseaux sociaux	998

- 17) Quels sont les 5 magasins avec le plus de feedbacks ?

```
1 • SELECT
2     RM.libelle_de_commune AS libelle_commune,
3     COUNT(*) AS nombre_feedbacks
4 FROM
5     retour_client RC
6 JOIN
7     Ref_magasin RM ON RC.ref_magasin = RM.ref_magasin
8 GROUP BY
9     RM.libelle_de_commune
10 ORDER BY
11     nombre_feedbacks DESC
12 LIMIT 5;
```

Result Grid		Filter Rows:
	libelle_commune	nombre_feedbacks
▶	Mareuil-les-Meaux	55
	Osny	49
	Lognes	47
	Villecresnes	45
	Ivry-sur-Seine	44

4) Requêtes SQL et Analyses

- 1^{ère} recommandation : les 5 magasins avec les plus faibles scores NPS

```
1 • SELECT
2     RM.libelle_de_commune AS Magasin,
3     ROUND((SUM(CASE WHEN RC.recommandation = 0 OR RC.recommandation = '' THEN 1 ELSE 0 END) / COUNT(*) * 100), 2) AS Taux_Recommandation_Faible
4 FROM
5     retour_client RC
6 JOIN
7     ref_magasin RM ON RC.ref_magasin = RM.ref_magasin
8 GROUP BY
9     RM.libelle_de_commune
10 ORDER BY
11     Taux_Recommandation_Faible DESC
12 LIMIT 5;
13
```

Result Grid | Filter Rows:

	Magasin	NPS
▶	Nanterre	6.25
	Buchelay	8.82
	Aubervilliers	14.29
	Fontenay-sous-Bois	15.38
	Lognes	17.02

- 2^{ème} recommandation : les 5 magasins avec les plus faibles recommandations

```
1 • SELECT
2     RM.libelle_de_commune AS Magasin,
3     ROUND((SUM(CASE WHEN RC.recommandation = 0 OR RC.recommandation = '' THEN 1 ELSE 0 END) / COUNT(*) * 100), 2) AS Taux_Recommandation_Faible
4 FROM
5     retour_client RC
6 JOIN
7     ref_magasin RM ON RC.ref_magasin = RM.ref_magasin
8 GROUP BY
9     RM.libelle_de_commune
10 ORDER BY
11     Taux_Recommandation_Faible DESC
12 LIMIT 5;
13
```

Result Grid | Filter Rows:

	Magasin	Taux_Recommandation_Faible
▶	Montgeron	47.37
	Herblay	45.00
	Carrières-sur-Seine	41.94
	Nanterre	40.63
	Saint-Denis	40.54

5) Cohérence des données



❖ Champ « typologie_produit » :

La donnée fournie sur le dictionnaire des données et sur le schéma de la BDD pour le champ typologie_produit était différente (Int sur le dictionnaire des données et char sur la BDD : nous avons conservé le format Char avec une taille à 50.

❖ Taille du champ « recommandation » :

La taille n'est pas précisée dans le dictionnaire des données. Nous l'avons mise à 50 dans le tableau « retour_client ».

❖ Clé étrangère :

La clef étrangère n'est pas précisée dans le dictionnaire des données (cle_produit) dans le tableau retour_client.

❖ Colonne « recommandations » :

Il y a des lignes vides dans la colonne « Recommandations » du tableau « retour_client ». Nous avons considéré que le client ne s'est pas prononcé tout en maintenant le total des lignes vides pour le calcul du pourcentage de recommandation.

POUR ALLER PLUS LOIN



SE COMPARER AUX AUTRES ENSEIGNES CONCURRENTIELLES EN TERME DE SATISFACTION CLIENT (ETUDES NIELSEN...)



ENTAMER UNE DEMARCHE DE CERTIFICATION DE SERVICE GRANDE DISTRIBUTION PAR L'AFNOR



ORGANISER DES TABLES RONDES CLIENTS POUR DES ENQUETES QUALITATIVES





Merci pour votre attention