Présentation du projet :

- ***Un rappel du contexte ;***
- ***Les bonnes pratiques en termes de stratégie de sauvegarde, de stockage et d'accès aux données de notre base de données ;***
- ***La méthodologie d'extraction et les modifications que tu as faites sur la base de données avec des captures d'écran :***
    - ***De la base de données extraite,***
    - ***Du dictionnaire de données mis à jour avec l'ajout des champs de la nouvelle table,***
    - ***Du nouveau schéma relationnel,***
    - ***Du système de gestion de base de données avec la nouvelle table ;***
- ***Les requêtes SQL avec leurs résultats ;***
- ***Le contrôle de la cohérence des données.***

1) Un rappel du contexte :

Dans le cadre de l'amélioration du suivi et de l'analyse des retours clients, une base de données a été mise en place pour centraliser les informations collectées. Cependant, après une première analyse, il apparaît que cette base ne permet pas de répondre à l'ensemble des besoins exprimés par le service client. Une mise à jour et une structuration des données sont donc nécessaires afin d'optimiser leur exploitation.

L'objectif final est de fournir un rapport clair et concis permettant au service client de mieux exploiter les données et d'optimiser l'analyse des retours clients.

2) Les bonnes pratiques en termes de stratégie de sauvegarde, de stockage et d'accès aux données de notre base de données :

❑ ***ACCES :***

- Filtrer les données sensibles et personnaliser l'affichage en utilisant des fonctions telles que `SUBSTRING` pour obscurcir les informations comme les adresses email et les numéros de téléphone.
    - ♣ Dans ce cas, pas besoin car pas de données sensibles
- Limitez l'accès aux données en veillant à ce que seuls les utilisateurs autorisés puissent y accéder, en utilisant des rôles et des permissions.
- Tenez un registre des accès aux données pour savoir qui a accès à quoi, et si des modifications sont nécessaires.
- *Authentification sécurisée* : Utiliser des mots de passe forts et activer l'authentification à deux facteurs si possible.
- *Chiffrement des données* : Mettre en place le chiffrement (SSL/TLS) pour sécuriser les échanges entre le client et le serveur MySQL.

- *Limiter les connexions simultanées* : Définir une limite sur le nombre de connexions (max_connections) pour éviter une surcharge du serveur.

❑ ***STOCKAGE :***

- *Indexation optimisée* : Utiliser des index sur les colonnes souvent utilisées dans les requêtes pour améliorer les performances.
- *Partitionnement des tables* : Diviser les grandes tables en partitions pour accélérer les requêtes et optimiser le stockage.
- *Archivage des anciennes données* : Déplacer les données historiques vers une base séparée ou un entrepôt de données pour alléger la base opérationnelle.
- *Surveillance de l'espace disque* : Configurer des alertes pour éviter une saturation du stockage.
- Appeler ces fonctions dans des requêtes SQL pour centraliser et simplifier la logique métier.

❑ ***SAUVEGARDE :***

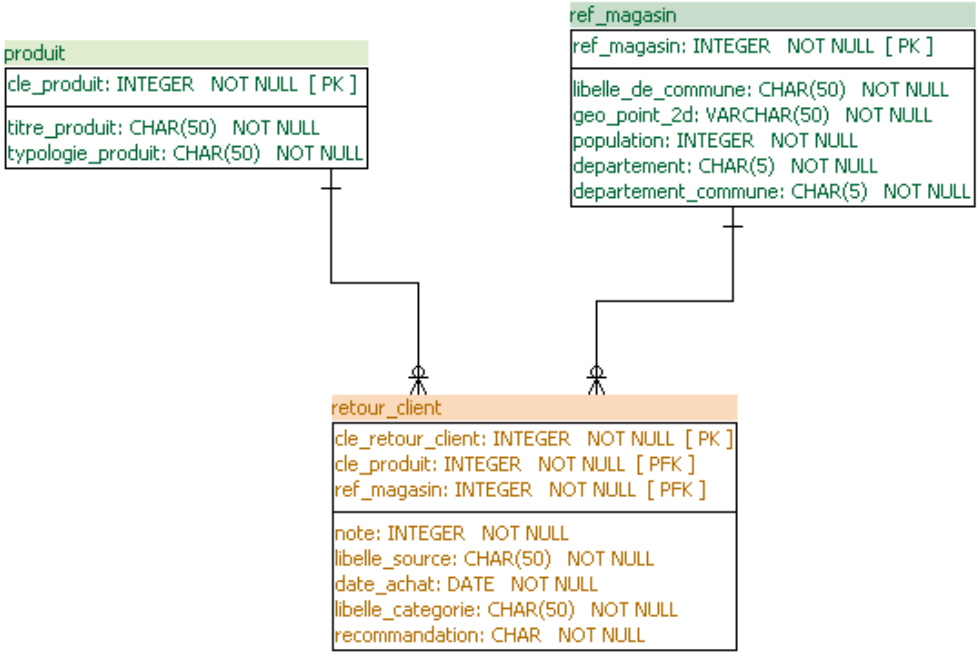
- *Fréquence des sauvegardes* : Il est important de définir à quelle fréquence vous effectuerez des sauvegardes (quotidiennes, hebdomadaires, etc.) en fonction de la criticité des données.
- *Sauvegardes incrémentales* : Pour optimiser l'espace, stocker uniquement les changements effectués après la dernière sauvegarde complète.
- *Réplication des bases* : Utiliser un serveur secondaire (réplication MySQL) pour assurer une continuité de service en cas de panne.
- *Stockage hors site* : Sauvegarder les fichiers sur un serveur distant ou sur le cloud pour éviter la perte de données en cas de problème matériel.
- *Tests de restauration* : Vérifier régulièrement que les sauvegardes sont exploitables et restaurables en cas de besoin.

3) La méthodologie d'extraction et les modifications que tu as faites sur la base de données avec des captures d'écran :

✓ **Dictionnaires des données mis à jour avec l'ajout des champs 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 des magasin
	departement	CHAR	5		numéro de département
	departement_commune	CHAR	5		numéro de commune
	libelle_de_commune	CHAR	50		nom de la commune
	geo_point_2d	VARCHAR	50		point géographique - lat long
	population	INT			nombre d'habitants

✓ Nouveau schéma relationnel :

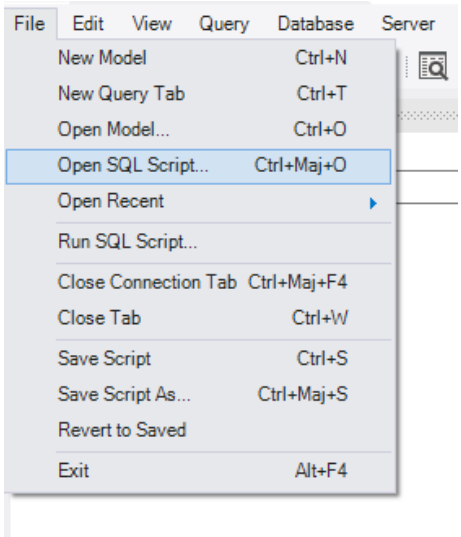


```sql
## Création de la table Régions :

DROP TABLE IF EXISTS Produit;
CREATE TABLE Produit (
    cle_produit INT PRIMARY KEY,
    titre_produit CHAR(50),
    typologie_produit INT
);

DROP TABLE IF EXISTS Retour_client;
CREATE TABLE Retour_client (
    cle_retour_client INT PRIMARY KEY,
    note INT,
    cle_produit INT,
    ref_magasin INT,
    date_achat DATE,
    libelle_source CHAR(50),
    libelle_categorie CHAR(50),
    recommandation CHAR,
    FOREIGN KEY (cle_produit) REFERENCES Produits(cle_produit)
);
```

✓ **Système de gestion de base de données avec la nouvelle table :**

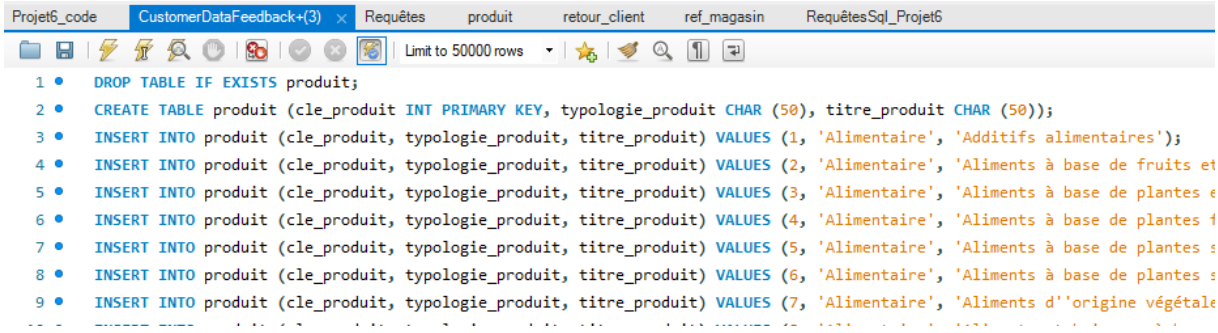
Étapes pour importer un fichier .sql dans MySQL :

1. Lancez l'application MySQL Workbench.
2. Aller sur File et importer le code de la création des tables :



3. Sélectionner la base de données.
4. Exécuter le script :
    a. Une fois le fichier chargé dans l'éditeur, vous verrez le contenu du script SQL.

b. Cliquez sur le bouton **Execute SQL** (Exécuter SQL) ou utilisez le raccourci `F5` pour exécuter toutes les instructions contenues dans le fichier.

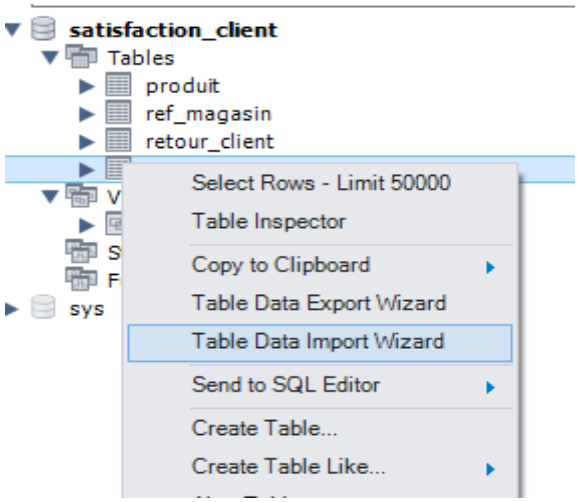


5. Base de données mise à jour avec le fichier ref_magasin pour créer une nouvelle table :

- Lors de la création de la nouvelle table dans la base de données, le format de cette table doit correspondre à celui spécifié dans le dictionnaire.
- L'encodage lors de l'importation du fichier doit être en UTF-8 pour gérer les caractères spéciaux présents.
- Création de la table ref_magasin :

```
30  DROP TABLE IF EXISTS ref_magasin;
31  CREATE TABLE ref_magasin (
32      ref_magasin INT PRIMARY KEY,
33      departement VARCHAR(50),
34      departement_commune VARCHAR(50),
35      libelle_de_commune VARCHAR(50),
36      population INT,
37      geo_point_2d VARCHAR(50)
38  );
39
40  select * From ref_magasin;
```

Importation de la table ref_magasin :


satisfaction_client
Tables
produit
ref_magasin
retour_client
Select Rows - Limit 50000
Table Inspector
Copy to Clipboard
Table Data Export Wizard
Table Data Import Wizard
Send to SQL Editor
Create Table...
Create Table Like...
sys

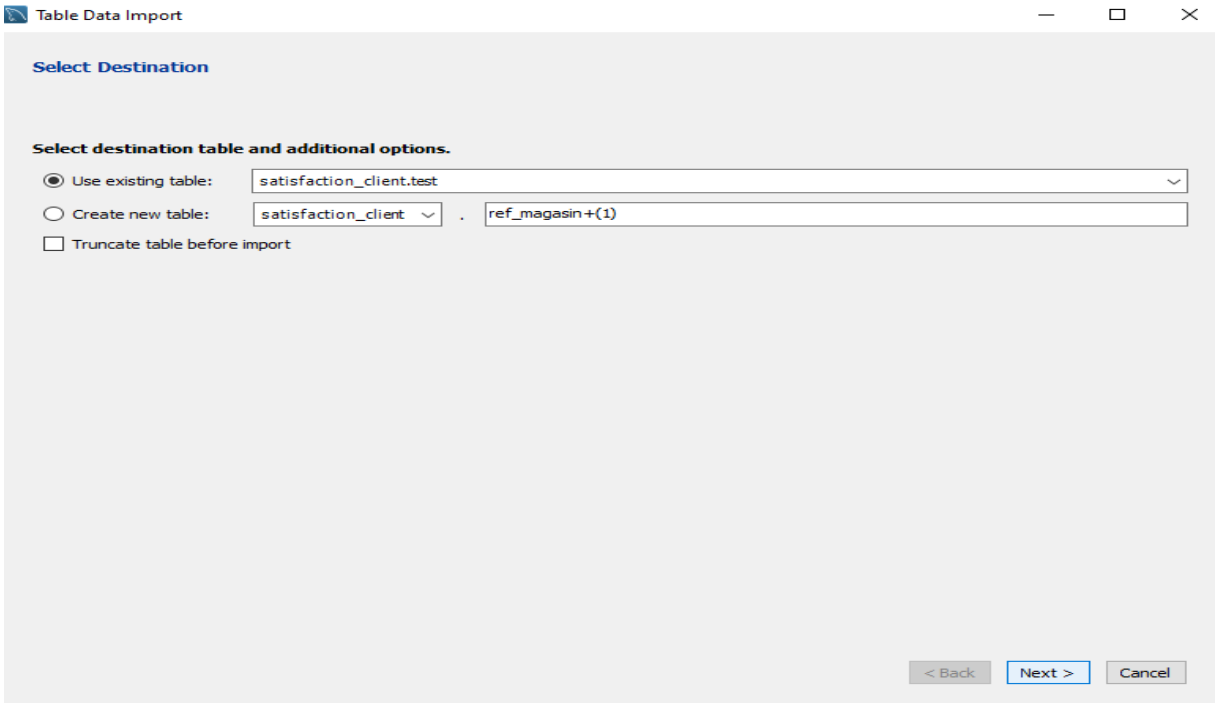


Table Data Import
Select Destination
Select destination table and additional options.
Use existing table:
satisfaction_client.test
Create new table:
satisfaction_client
ref_magasin+(1)
Truncate table before import
< Back
Next >
Cancel

Projet6_code | CustomerDataFeedback+(3) | Requêtes | produit | retour_client | ref_magasin | RequêtesSql_Projet6

Limit to 50000 rows

```
1 •  SELECT * FROM satisfaction_client.ref_magasin;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

ref_magasin	departement	departement_commune	libelle_de_commune	population
1	95	95355	Magny-en-Vexin	5752
2	92	92012	Boulogne-Billancourt	114205
3	75	75105	Paris 5e Arrondissement	60938
4	94	94028	Créteil	89985
5	94	94075	Villecresnes	9598
6	95	95476	Osny	16236
7	75	75120	Paris 20e Arrondissement	196880
8	93	93001	Aubervilliers	76087
9	93	93070	Saint-Ouen	47189

La table est bien chargée.

6. Vérification doublons/ null :

Doublons absolus

SELECT COUNT(*) AS nbr_doublons, cle_produit, produit.titre_produit, produit.typologie_produit
FROM produit
GROUP BY cle_produit, produit.titre_produit, produit.typologie_produit
HAVING COUNT(*) > 1;

Idem pour les autres tables

Lignes NULL

SELECT *
FROM retour_client
WHERE retour_client.recommandation IS NULL
OR retour_client.recommandation = '';

4) Les requêtes SQL avec leurs résultats :

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

```
/* Q1 : Quel est le nombre de retours clients sur la livraison ? */
SELECT COUNT(*) AS nombre_retours_sur_livraison
FROM retour_client
WHERE libelle_categorie = 'livraison';
```

nombre_retours_sur_livraison
639

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

```sql
/* Q2 : Quelle est la liste des notes des clients sur les réseaux sociaux sur les TV ? */
SELECT note
FROM retour_client
JOIN produit ON retour_client.cle_produit = produit.cle_produit
WHERE retour_client.libelle_source = 'réseaux sociaux'
AND produit.titre_produit = 'TV'
GROUP BY note ;
```

	note
▶	10
	9
	8

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

```sql
/* Q3 : Quelle est la note moyenne pour chaque catégorie de produit ? (Classé de la meilleure à la moins bonne) */
SELECT produit.typologie_produit AS categorie_de_produit, ROUND (AVG (retour_client.note),1) AS note_moyenne
FROM retour_client
JOIN produit ON retour_client.cle_produit = produit.cle_produit
GROUP BY produit.typologie_produit
ORDER BY note_moyenne DESC;
```

	categorie_de_produit	note_moyenne
▶	High-Tech	8.2
	Loisirs	8.1
	Alimentaire	8.0
	Maison	7.9

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

```sql
/* Q4 : Quels sont les 5 magasins avec les meilleures notes moyennes ? */
SELECT ref_magasin,ROUND (AVG (retour_client.note),1) AS note_moyenne
FROM retour_client
GROUP BY ref_magasin
ORDER BY note_moyenne DESC
LIMIT 5;
```

	ref_magasin	note_moyenne
▶	75	8.7
	23	8.5
	19	8.5
	62	8.5
	78	8.5

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

```
/* Q5 : Quels sont les magasins qui ont plus de 12 feedbacks sur le drive ? */
SELECT ref_magasin, COUNT(retour_client.note) AS nombre_de_feedback
FROM retour_client
WHERE libelle_categorie = 'drive'
GROUP BY ref_magasin
HAVING COUNT(retour_client.note) > 12;
```

	ref_magasin	nombre_de_feedback
▶	67	14
	63	13
	45	13

6)Quel est le classement des départements par note ?

```
/* Q6 : Quel est le classement des départements par note ? */
SELECT ref_magasin.departement, ROUND (AVG(retour_client.note),2) AS note_moyenne
FROM retour_client
JOIN ref_magasin ON retour_client.ref_magasin = ref_magasin.ref_magasin
GROUP BY ref_magasin.departement
ORDER BY note_moyenne DESC;
```

	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

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

```
/* Q7 : Quelle est la typologie de produit qui apporte le meilleur service après-vente ? */
SELECT produit.typologie_produit, ROUND (AVG (retour_client.note),2) AS recommandation
FROM produit
JOIN retour_client ON retour_client.cle_produit = produit.cle_produit
WHERE retour_client.libelle_categorie = 'service après-vente'
GROUP BY produit.typologie_produit
ORDER BY recommandation DESC
LIMIT 1 ;
```

	typologie_produit	recommandation
▶	Loisirs	8.51

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

```
/* Q8 : Quelle est la note moyenne sur l'ensemble des boissons ? */
SELECT ROUND(AVG(retour_client.note), 2) AS note_moyenne
FROM produit
JOIN retour_client ON retour_client.cle_produit = produit.cle_produit
WHERE titre_produit LIKE 'Boisson%';

SELECT ROUND(AVG(retour_client.note), 2) AS note_moyenne
FROM produit
JOIN retour_client USING (cle_produit)
WHERE titre_produit LIKE 'Boisson%';
```

	note_moyenne
▶	8.32

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

```
/* Q9 : Quel est le classement des jours de la semaine où l'expérience client est la meilleure expérience en magasin ? */
SELECT DAYOFWEEK(retour_client.date_achat) AS jour_semaine,
ROUND(AVG(retour_client.note), 2) AS note_moyenne
FROM retour_client
WHERE retour_client.libelle_categorie LIKE '%magasin%'
GROUP BY jour_semaine
ORDER BY jour_semaine;
```

	jour_semaine	note_moyenne
▶	1	8.18
	2	7.74
	3	7.95
	4	7.99
	5	8.04
	6	8.07
	7	8.34

1 => Dimanche ....

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

```sql
/* Q10 : Sur quel mois a-t-on le plus de retour sur le service après-vente ? */
SELECT MONTH(retour_client.date_achat) AS mois  , COUNT(retour_client.note) AS nombre_retours
FROM retour_client
WHERE retour_client.libelle_categorie LIKE '%ap%vente'
GROUP BY mois
ORDER BY nombre_retours DESC
LIMIT 1;
```

	nombre_retours	mois
▶	55	10

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)

```sql
/* Q11 : 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 = 1 or
recommandation = 0) as "% de recommandation"
from retour_client
where recommandation = 1;
```

	% de recommandation
▶	70.5000

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

```sql
/* Q12 : Quels sont les magasins qui ont une note inférieure à la moyenne ? */
WITH note_moyenne_totale AS (
    SELECT AVG(note) AS moyenne_globale FROM retour_client
)
SELECT retour_client.ref_magasin,
       ROUND(AVG(retour_client.note), 2) AS note_moyenne_mag
FROM retour_client
GROUP BY retour_client.ref_magasin
HAVING note_moyenne_mag < (SELECT moyenne_globale FROM note_moyenne_totale)
ORDER BY note_moyenne_mag DESC;
```

ref_magasin	note_moyenne_mag
29	8.05
50	8.05
36	8.05
3	8.04
47	8.03
7	8.00
59	8.00
38	7.97
53	7.97
73	7.97
79	7.95
64	7.93
45	7.93
13	7.92
6	7.90
58	7.90
14	7.90
34	7.89
33	7.89
20	7.87
63	7.86
65	7.85
54	7.84

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

```sql
/* Q13 : Quelles sont les typologies produits qui ont amélioré leur moyenne entre
le 1er et le 2ème trimestre 2021 ? */
WITH
t1 as
(SELECT typologie_produit, ROUND(AVG(retour_client.note),2) AS moyenne_t1
FROM produit
JOIN retour_client ON retour_client.cle_produit = produit.cle_produit
WHERE date_achat BETWEEN '2021-01-01' AND '2021-03-31'
GROUP BY typologie_produit),
t2 as
(SELECT typologie_produit, ROUND(AVG(retour_client.note),2) AS moyenne_t2
FROM produit
JOIN retour_client ON retour_client.cle_produit = produit.cle_produit
WHERE date_achat BETWEEN '2021-04-01' AND '2021-06-30'
GROUP BY typologie_produit)

SELECT typologie_produit, moyenne_t1, moyenne_t2, ROUND(((moyenne_t2-moyenne_t1)/moyenne_t1*100),2) AS amelioration_pourcentage
FROM t1
Join t2 using (typologie_produit)
WHERE ((moyenne_t2-moyenne_t1)/moyenne_t1)>=0 ;
```

typologie_produit	moyenne_t1	moyenne_t2	amelioration_pourcentage
Alimentaire	7.99	8.06	0.88
Loisirs	8.00	8.34	4.25

Question 14 : NPS

1. Common Table Expression (CTE) nps_categorie :

    a. Cette partie catégorise les notes des clients en trois groupes :
        i. 'promoteur' pour les notes de 9 et 10
        ii. 'detracteur' pour les notes de 0 à 6
        iii. 'passif' pour les notes de 7 et 8
    b. Elle compte ensuite le nombre total de réponses dans chaque catégorie
2. Requête principale :
    a. Calcule le pourcentage de promoteurs :
       SUM(CASE WHEN categorie = 'promoteur' THEN total ELSE 0 END)*100.0/SUM(total)
    b. Calcule le pourcentage de détracteurs :
       SUM(CASE WHEN categorie = 'detracteur' THEN total ELSE 0 END)*100.0/SUM(total)
    c. Soustrait le pourcentage de détracteurs du pourcentage de promoteurs
    d. Arrondit le résultat à 2 décimales avec ROUND(..., 2)
3. Le résultat final :
    a. Donne le Net Promoter Score (NPS), qui est la différence entre le pourcentage de promoteurs et le pourcentage de détracteurs
    b. Le NPS peut varier de -100 (si tous sont détracteurs) à +100 (si tous sont promoteurs)

```sql
/* Q14 : NPS */
WITH nps_categorie AS (
SELECT
CASE
      WHEN retour_client.note >= 9 THEN 'promoteur'
      WHEN retour_client.note <= 6 THEN 'detracteur'
      ELSE 'passif'
      END AS categorie,
      COUNT(*) AS total
FROM retour_client
GROUP BY
CASE
      WHEN retour_client.note >= 9 THEN 'promoteur'
      WHEN retour_client.note <= 6 THEN 'detracteur'
      ELSE 'passif'
END)
SELECT ROUND(
(SUM(CASE WHEN categorie = 'promoteur' THEN total ELSE 0 END)*100.0/SUM(total))-
(SUM(CASE WHEN categorie = 'detracteur' THEN total ELSE 0 END)*100.0/SUM(total))) AS nps
FROM nps_categorie;
```

	nps
▶	31

15)NPS par source

```sql
/* Q15 : NPS Source*/
CREATE VIEW categorie_nps_source_vw AS
SELECT retour_client.libelle_source,
CASE
   WHEN retour_client.note >= 9 THEN 'promoteur'
   WHEN retour_client.note <= 6 THEN 'detracteur'
   ELSE 'passif'
   END AS categorie,
   COUNT(*) AS total
FROM retour_client
GROUP BY retour_client.libelle_source,
CASE
   WHEN retour_client.note >= 9 THEN 'promoteur'
   WHEN retour_client.note <= 6 THEN 'detracteur'
   ELSE 'passif'
END;
SELECT libelle_source, ROUND(
(SUM(CASE WHEN categorie = 'promoteur' THEN total ELSE 0 END)*100.0/SUM(total))-
(SUM(CASE WHEN categorie = 'detracteur' THEN total ELSE 0 END)*100.0/SUM(total))) AS nps
FROM categorie_nps_source_vw
GROUP BY libelle_source
ORDER BY nps;
```

libelle_source	nps
email	30
réseaux sociaux	30
téléphone	34

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

```sql
/* Q16 : Exemple : Quel est le nombre de retour clients par source ? */
SELECT retour_client.libelle_source, COUNT(*) AS nombre_retours
FROM retour_client
GROUP BY retour_client.libelle_source
ORDER BY retour_client.libelle_source DESC;
```

	libelle_source	nombre_retours
▶	téléphone	970
	réseaux sociaux	998
	email	1032

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

```sql
/* Q17 : Exemple 2 : Quels sont les 5 magasins avec le plus de feedbacks ? */
SELECT ref_magasin.ref_magasin, COUNT(*) as nombre_feedback
FROM ref_magasin
JOIN retour_client ON retour_client.ref_magasin = ref_magasin.ref_magasin
GROUP BY ref_magasin.ref_magasin
ORDER BY nombre_feedback DESC
LIMIT 5;
```

	ref_magasin	nombre_feedback
▶	29	55
	6	49
	80	47
	5	45
	63	44