

RetailInsight360 Antoine Vatin 22/01/2025

## 1) Contexte et expression du besoin



### **Enjeux**:

- Améliorer la satisfaction client pour renforcer leur fidélité.
- Exploiter les données collectées pour orienter les décisions stratégiques.
- Valoriser l'image de marque de BestMarket comme une entreprise à l'écoute de ses clients.

## **©** Objectif final:

Transformer les feedbacks en actions concrètes pour une meilleure expérience en magasin et en ligne.

## 2) Sauvegarde et stockage de la BDD

- Automatisation des sauvegardes
  - Protection continue des données.
  - Stockage hors site pour assurer la récupération en cas de sinistre.
- Contrôle d'accès strict
- Gestion des permissions par rôles pour limiter l'accès aux données sensibles.
- Réduction des risques de fuites et renforcement de la sécurité.
- **©** Objectif:

• Assurer l'intégrité, la confidentialité et la disponibilité des données.

## 3) Méthodologie suivie

## **©** Objectif:

• Une base de données structurée et fiable pour des analyses précises.

## tapes du processus

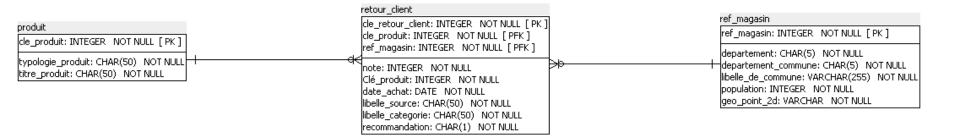
- 1. Vérification et nettoyage des données pour assurer leur qualité.
- 2. Importation des données via fichiers CSV dans PGAdmin.
- 3. Adaptation du schéma relationnel et du dictionnaire de données pour garantir cohérence et compatibilité.
- Mise à jour et exploitation à l'aide d'outils PGAdmin.



## Dictionnaire de données

|   | Nom du champs       | Type de données | Taille | Contrainte   | Description  |
|---|---------------------|-----------------|--------|--------------|--|
|   | cle_retour_client   | INT             |        | Clé primaire | ID unique pour les retours clients   |
|   |                     |                 |        |              | 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 |
| nt  | note                | INT             |        |              | probabilité que vous recommandiez notre entreprise à votre entourage ?"  |
| e<br>:liei                                  | Clé_produit         | INT             |        |              | ID des produits  |
|   | ref_magasin         | INT             |        |              | ID des magasins  |
| T<br>eto                                    | date_achat          | DATE            |        |              | Date à laquelle l'achat du client a eu lieu  |
| 8   | 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            | 1      |              | Recommandation laissée par le client à la question 'Recommandez vous l'entreprise?' True / False                                   |
| e<br>Jit                                    | cle_produit         | INT             |        | Clé primaire | ID unique pour les produits  |
| npo.  | titre_produit       | CHAR            | 50     |              | Libellé des produits   |
| Table Table Table Table Table Table Froduit | typologie_produit   | CHAR            | 50     |              | Typologie des produits (Alimentaire, High-tech etc)  |
|   | ref_magasin         | INT             |        | Clé primaire | ID unique pour les magasin   |
| ısin  | departement         | CHAR            | 5      |              | Numéro du département  |
| ble<br>aga                                  | departement_commune | CHAR            | 5      |              | Code de la commune   |
|   | libelle_de_commune  | VARCHAR         | 255    |              | Nom de la commune  |
| Ref   | population          | INT             |        |              | Population de la commune   |
|   | geo_point_2d        | VARCHAR         |        |              | Coordonnées géographiques  |

### Schéma relationnel



## 4) Requêtes SQL et Analyses

### **Volume des Retours Clients**

Quel est le nombre de retours clients sur la livraison ?

#### SELECT

 $\hbox{\it COUNT(*) AS nombre\_retours\_livraison $$--$ Compte le nombre total de retours clients liés à la livraison $$FROM retour\_client$$ 

WHERE libelle\_categorie = 'livraison'; -- Filtre uniquement les retours de la catégorie "livraison"

|   | nombre_retours_livraison bigint |
|---|---------------------------------|
| 1 | 639                             |

Quel est le nombre de retours clients par source ?

#### SELECT

libelle\_source AS source, -- Regroupe les retours clients par canal d'origine
COUNT(\*) AS nombre\_retours -- Compte le nombre total de retours par source
FROM retour\_client
GROUP BY libelle\_source -- Regroupe les données par type de source

ORDER BY nombre\_retours DESC; -- Trie les résultats du plus grand au plus petit nombre de retours

source

nombre\_retours

|   | character (50)  | nombre_retours<br>bigint |
|---|-----------------|--------------------------|
| 1 | email           | 1032                     |
| 2 | réseaux sociaux | 998                      |
| 3 | téléphone       | 970                      |

### **Analyse des Notes Clients**

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

Quelle est la note moyenne pour chaque catégorie de produit ?

```
SELECT DISTINCT

rc.note, -- Note attribuée par le client
p.titre_produit -- Nom du produit évalué

FROM retour_client rc

JOIN produit p

ON rc.cle_produit = p.cle_produit -- Jointure entre retours clients et produits

WHERE rc.libelle_source = 'réseaux sociaux' -- Filtre sur les retours issus des réseaux sociaux

AND p.titre_produit = 'TV'; -- Filtre uniquement les produits "TV"
```

|   | note<br>integer | titre_produit<br>character (50) |
|---|-----------------|---------------------------------|
| 1 | 8               | TV                              |
| 2 | 9               | TV                              |
| 3 | 10              | TV                              |

p.typologie\_produit, -- Catégorie de produit

ROUND(AVG(rc.note), 2) AS note\_moyenne -- Moyenne des notes clients, arrondie à 2 décimales

FROM retour\_client rc

JOIN produit p

ON rc.cle\_produit = p.cle\_produit -- Jointure avec la table produit pour récupérer la catégorie

GROUP BY p.typologie\_produit -- Regroupe par catégorie de produit

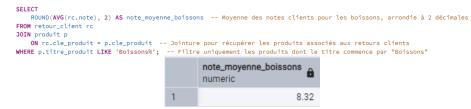
ORDER BY note\_moyenne DESC; -- Trie les résultats du meilleur au moins bon

| , |   |                                     |   |                      |  |  |  |
|---|---|-------------------------------------|---|----------------------|--|--|--|
|   |   | typologie_produit<br>character (50) | â | note_moyenne numeric |  |  |  |
|   | 1 | High-Tech                           |   | 8.16                 |  |  |  |
|   | 2 | Loisirs                             |   | 8.09                 |  |  |  |
|   | 3 | Alimentaire                         |   | 8.04                 |  |  |  |
|   | 4 | Maison                              |   | 7.85                 |  |  |  |

### **Analyse des Notes Clients**

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

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



```
SELECT

p.typologie_produit, -- Catégorie de produit

ROUND(AVG(rc.note), 2) AS note_moyenne_sav -- Moyenne des notes clients pour le service après-vente, arrondie à 2 décimales

FROM retour_client rc

JOIN produit p

ON rc.cle_produit = p.cle_produit -- Association des retours clients avec les produits

WHERE rc.libelle_categorie = 'service après-vente' -- Filtrage sur la catégorie "service après-vente"

GROUP BY p.typologie_produit -- Regroupement par typologie de produit

ORDER BY note_moyenne_sav DESC -- Trie les résultats du meilleur au moins bon

LIMIT 1; -- Sélectionne uniquement la typologie de produit la mieux notée

typologie_produit

Character (50)

1 Loisirs

8.51
```

### Performance des Magasins

Quels sont les 5 magasins avec les meilleures notes moyennes ?

```
SELECT

rc.ref_magasin, -- Identifiant du magasin

ROUND(AVG(rc.note), 2) AS note_moyenne -- Moyenne des notes clients, arrondie à 2 décimales

FROM retour_client rc

GROUP BY rc.ref_magasin -- Regroupement par magasin

ORDER BY note_moyenne DESC -- Trie les magasins du mieux noté au moins bien noté

LINIT 5; -- Sélectionne les 5 meilleurs magasins
```

|   | ref_magasin<br>integer | note_moyenne numeric |
|---|------------------------|----------------------|
| 1 | 75                     | 8.73                 |
| 2 | 78                     | 8.55                 |
| 3 | 62                     | 8.50                 |
| 4 | 23                     | 8.48                 |
| 5 | 19                     | 8.45                 |

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

```
WITH moyenne_globale AS
   -- Calcul de la note moyenne globale de tous les magasins
   SELECT AVG(note) AS note_movenne_globale
   FROM retour client
SELECT
   rc.ref_magasin, -- Identifiant du magasin
   ROUND(AVG(rc.note), 2) AS note_moyenne_magasin -- Moyenne des notes du magasin, arrondie à 2 décimales
FROM retour_client rc
GROUP BY rc.ref_magasin -- Regroupement par magasin
HAVING AVG(rc.note) < (SELECT note movenne globale FROM movenne globale) -- Sélection des magasins sous la movenne globale
ORDER BY note_moyenne_magasin ASC; -- Trie du moins bien noté au plus proche de la moyenne
                                                             ref_magasin
                                                                              note_moyenne_magasin
                        note_moyenne_magasin
        ref_magasin
                                                              integer
                                                                               numeric
                         numeric
                                                                           20
                                                                                                     7.87
                                               7.38
                                                                           34
                                                                                                     7.89
                    81
                                               7.44
                                                                           33
                                                                                                     7.89
3
                     82
                                               7.53
                                                                                                     7.90
                                                                           6
                     46
                                               7.56
                                                                           14
                                                                                                     7.90
5
                    55
                                               7.59
                                                                           58
                                                                                                     7.90
6
                    24
                                               7.62
                                                                           13
                                                                                                     7.92
                     80
                                               7.62
                                                                           64
                                                                                                     7.93
8
                                               7.66
                                                                           45
                                                      29
                                                                                                     7.93
9
                     44
                                               7.67
                                                                           79
                                                                                                     7.95
10
                    74
                                               7.70
                                                                           38
                                                                                                     7.97
11
                     57
                                               7.73
                                                                           73
                                                                                                     7.97
12
                     76
                                               7.74
                                                      33
                                                                           53
                                                                                                     7.97
13
                     68
                                               7.79
                                                                           59
                                                                                                     8.00
14
                                               7.82
                                                                                                     8.00
15
                                               7.83
                     18
                                                                           47
                                                                                                     8.03
16
                     25
                                               7.83
                                                                           3
                                                                                                     8.04
17
                                               7.84
                                                                           50
                                                                                                     8.05
18
                    51
                                               7.84
                                                                           36
19
                                                                                                     8.05
                     65
                                               7.85
```

7.86

20

63

### Performance des Magasins

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

SELECT

rc.ref\_magasin, -- Identifiant du magasin

COUNT(\*) AS nombre\_feedbacks -- Nombre total de feedbacks pour chaque magasin

FROM retour\_client rc

WHERE rc.libelle\_categorie = 'drive' -- Filtre uniquement les retours liés au Drive

GROUP BY rc.ref\_magasin -- Regroupe les résultats par magasin

HAVING COUNT(\*) > 12 -- Affiche uniquement les magasins ayant plus de 12 feedbacks

|   | ref_magasin<br>integer | nombre_feedbacks bigint |
|---|------------------------|-------------------------|
| 1 | 67                     | 14                      |
| 2 | 63                     | 13                      |
| 3 | 45                     | 13                      |

ORDER BY nombre feedbacks DESC: -- Trie les résultats par nombre de feedbacks en ordre décroissant

Quels sont les 5 magasins avec le plus de feedbacks?

```
SELECT

ref_magasin AS magasin, -- Identifiant du magasin

COUNT(*) AS nombre_feedbacks -- Nombre total de retours clients pour chaque magasin

FROM retour_client

GROUP BY ref_magasin -- Regroupement par magasin

ORDER BY nombre_feedbacks DESC -- Trie du plus grand nombre de feedbacks au plus petit

LIMIT 5: -- Sélectionne uniquement les 5 magasins ayant reçu le plus de feedbacks
```

| 1 | 29 | 55 | 55 | 2 | 6 | 49 | 47 | 4 | 5 | 45 | 5 | 83 | 44 |

### **Analyse Géographique**

#### Quel est le classement des départements par note ?

```
SELECT

rm.departement AS departement, -- Code du département

ROUND(AVG(rc.note), 2) AS note_moyenne -- Moyenne des notes des retours clients, arrondie à 2 décimales

FROM retour_client rc

JOIN ref_magasin rm

ON rc.ref_magasin = rm.ref_magasin -- Association des retours clients avec les magasins et leurs départements

GROUP BY rm.departement -- Regroupement des résultats par département

ORDER BY note_moyenne DESC; -- Trie les départements du mieux noté au moins bien noté
```

|   | departement integer | note_moyenne numeric |
|---|---------------------|----------------------|
| 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                 |

## Évolution des Retours Clients dans le Temps

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

SELECT

TO\_CHAR(rc.date\_achat, 'Day') AS jour\_semaine, -- Extraction du jour de la semaine
ROUND(AVG(rc.note), 2) AS note\_moyenne -- Moyenne des notes clients, arrondie à 2 décimales
FROM retour\_client rc
WHERE rc.libelle\_categorie = 'expérience en magasin' -- Filtrage sur l'expérience en magasin
GROUP BY TO\_CHAR(rc.date\_achat, 'Day') -- Regroupement par jour de la semaine
ORDER BY note\_moyenne DESC; -- Tri des jours selon la satisfaction décroissante

|   | jour_semaine text | note_moyenne<br>numeric |
|---|-------------------|-------------------------|
| 1 | Saturday          | 8.34                    |
| 2 | Sunday            | 8.18                    |
| 3 | Friday            | 8.07                    |
| 4 | Thursday          | 8.04                    |
| 5 | Wednesday         | 7.99                    |
| 6 | Tuesday           | 7.95                    |
| 7 | Monday            | 7.74                    |

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

SELECT

TO\_CHAR(rc.date\_achat, 'Month') AS mois, -- Extraction du mois en toutes lettres

COUNT(\*) AS nombre\_retours -- Nombre total de retours clients pour chaque mois

FROM retour\_client rc

WHERE rc.libelle\_categorie = 'service après-vente' -- Filtrage sur la catégorie "service après-vente"

GROUP BY TO\_CHAR(rc.date\_achat, 'Month'), EXTRACT(MONTH FROM rc.date\_achat) -- Regroupement par mois avec tri correct

ORDER BY nombre\_retours DESC -- Trie les mois selon le nombre de retours en ordre décroissant

LIMIT 1; -- Sélectionne uniquement le mois avec le plus de retours

mois text nombre\_retours bigint

1 October 55

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

```
WITH movennes_par_typologie AS (
    -- Calcul de la moyenne des notes par typologie de produit et par trimestre
        p.typologie_produit,
            WHEN EXTRACT(QUARTER FROM rc.date_achat) = 1 THEN 'T1'
            WHEN EXTRACT(QUARTER FROM rc.date_achat) = 2 THEN 'T2'
        END AS trimestre.
        ROUND(AVG(rc.note), 2) AS movenne note -- Movenne des notes, arrondie à 2 décimales
    FROM retour_client rc
    JOIN produit p ON rc.cle produit = p.cle produit
    WHERE rc.date achat BETWEEN '2021-01-01' AND '2021-06-30' -- Filtre sur le premier semestre 2021
    GROUP BY p.typologie_produit, EXTRACT(QUARTER FROM rc.date_achat)
SELECT
    m1.typologie produit, -- Typologie de produit concernée
    m1.moyenne_note AS moyenne_T1, -- Note moyenne au 1er trimestre
    m2.moyenne_note AS moyenne_T2, -- Note moyenne au 2ème trimestre
    ROUND((m2.moyenne_note - m1.moyenne_note) * 100 / m1.moyenne_note, 2) AS "% évolution" -- Évolution en pourcentage
FROM movennes par typologie m1
JOIN moyennes_par_typologie m2
    ON m1.typologie_produit = m2.typologie_produit -- Jointure pour comparer les deux trimestres
WHERE m1.trimestre = 'T1'
  AND m2.trimestre = 'T2'
  AND m2.moyenne_note > m1.moyenne_note -- Sélectionne uniquement les typologies en amélioration
ORDER BY (m2.moyenne_note - m1.moyenne_note) DESC; -- Trie par la plus forte progression
```

|   | typologie_produit<br>character (50) € | moyenne_t1 numeric | moyenne_t2 numeric | % évolution numeric |
|---|---------------------------------------|--------------------|--------------------|---------------------|
| 1 | Loisirs                               | 8.00               | 8.34               | 4.25                |
| 2 | Alimentaire                           | 7.99               | 8.06               | 0.88                |

### **Net Promoter Score (NPS) & Satisfaction**

#### Quel est le pourcentage de recommandations client ?

```
SELECT

ROUND(

100 * COUNT(*) /

(SELECT COUNT(*) FROM retour_client WHERE recommandation IN ('1', '0')), 2

) AS "%_recommandation" -- Pourcentage des clients ayant recommandé l'entreprise

FROM retour_client

WHERE recommandation = '1'; -- Filtre les clients qui ont répondu "Oui" à la recommandation

mumeric

1 90.00
```

#### **NPS Global**

```
CREATE VIEW v_nps AS

SELECT

COUNT(*) AS total_feedbacks, -- Nombre total de feedbacks

COUNT(CASE WHEN note BETWEEN 9 AND 10 THEN 1 END) AS total_promoteurs,

COUNT(CASE WHEN note BETWEEN 0 AND 6 THEN 1 END) AS total_detracteurs,

ROUND(100.0 * COUNT(CASE WHEN note BETWEEN 9 AND 10 THEN 1 END) / COUNT(*), 2) AS pourcentage_promoteurs,

ROUND(100.0 * COUNT(CASE WHEN note BETWEEN 0 AND 6 THEN 1 END) / COUNT(*), 2) AS pourcentage_detracteurs,

ROUND(100.0 * (

(COUNT(CASE WHEN note BETWEEN 9 AND 10 THEN 1 END) -

COUNT(CASE WHEN note BETWEEN 0 AND 6 THEN 1 END))

) / COUNT(*), 2) AS nps -- Calcul du Net Promoter Score (NPS)

FROM retour_client;
```

# SELECT \* FROM v\_nps; -- Appel de la vue pour affichage total\_feedbacks bigint total\_detracteurs bigint bigint pourcentage\_promoteurs numeric pourcentage\_promoteurs numeric pourcentage\_detracteurs numeric n

#### NPS par source

```
CREATE VIEW v_nps_par_source AS
SELECT

rc.libelle_source AS source, -- Canal de retour client (ex: réseaux sociaux, email, téléphone)

COUNT(*) AS total_feedbacks, -- Nombre total de feedbacks par source

COUNT(CASE WHEN note BETWEEN 9 AND 10 THEN 1 END) AS total_promoteurs,

COUNT(CASE WHEN note BETWEEN 0 AND 6 THEN 1 END) AS total_detracteurs,

ROUND(100.0 * COUNT(CASE WHEN note BETWEEN 9 AND 10 THEN 1 END) / COUNT(*), 2) AS pourcentage_promoteurs,

ROUND(100.0 * COUNT(CASE WHEN note BETWEEN 0 AND 6 THEN 1 END) / COUNT(*), 2) AS pourcentage_detracteurs,

ROUND(100.0 * (

(COUNT(CASE WHEN note BETWEEN 9 AND 10 THEN 1 END) -

COUNT(CASE WHEN note BETWEEN 0 AND 6 THEN 1 END))

) / COUNT(*), 2) AS nps -- Calcul du Net Promoter Score (NPS)

FROM retour_client rc

GROUP BY rc.libelle_source -- Regroupe les résultats par canal de retour client

ORDER BY nps DESC;
```

SELECT \* FROM v\_nps\_par\_source; -- Appel de la vue pour affichage

|   | source character (50) | total_feedbacks<br>bigint | total_promoteurs bigint | total_detracteurs bigint | pourcentage_promoteurs numeric | pourcentage_detracteurs numeric | nps<br>numeric |
|---|-----------------------|---------------------------|-------------------------|--------------------------|--------------------------------|---------------------------------|----------------|
| 1 | téléphone             | 970                       | 403                     | 75                       | 41.55                          | 7.73                            | 33.81          |
| 2 | email                 | 1032                      | 387                     | 81                       | 37.50                          | 7.85                            | 29.65          |
| 3 | réseaux sociaux       | 998                       | 410                     | 115                      | 41.08                          | 11.52                           | 29.56          |

- Validation des Règles Métier
  - **Objectif**: Garantir que les données respectent les règles métier définies pour leur utilisation.
- Exemples de Règles Métier :
  - •Les notes clients doivent être comprises entre 0 et 10.
  - Les références des magasins doivent exister dans la table des magasins.
  - •Les dates d'achat ne doivent pas être dans le futur.



- Détection des Anomalies
- **Objectif**: S'assurer que les données respectent les règles métier et éviter les doublons dans les retours clients.
- Actions & Requêtes SQL pour identifier les anomalies :
  - Détecter les doublons dans les retours clients
  - Vérifier les retours avec des valeurs nulles ou incohérentes
  - Détecter les incohérences sur les dates de retour client

```
-- Détecter les doublons dans les retours clients

SELECT ref_magasin, cle_produit, date_achat, COUNT(*) AS occurrences

FROM retour_client

GROUP BY ref_magasin, cle_produit, date_achat

HAVING COUNT(*) > 1;
```

-- Vérifier les retours avec des valeurs nulles ou incohérentes SELECT \* FROM retour\_client WHERE not IS NULL OR ref\_magasin IS NULL OR cle\_produit IS NULL;

-- Détecter les incohérences sur les dates SELECT \* FROM retour\_client WHERE date\_achat > CURRENT\_DATE;

- ★ Mise en place d'un Système de Vérification Continue
  - Automatisation de la détection des doublons avec des triggers SQL.
  - Table de suivi des anomalies pour conserver un historique des corrections.
  - Vérification périodique des données via des scripts automatisés.
  - **Objectif final**: Assurer une base de données fiable et éviter les erreurs d'analyse causées par des données dupliquées ou incohérentes.



- Contrôle des correspondances entre tables
  - **Objectif**: Vérifier que les relations entre les tables respectent les contraintes relationnelles et assurer l'intégrité des données.
- Actions & Requêtes SQL contrôler l'intégrité référentielle :
  - Identifier les retours clients liés à un magasin inexistant
  - Détecter les produits associés à un retour client mais inexistants dans la table produit

```
-- Vérifier les retours clients liés à un magasin inexistant

SELECT rc.*

FROM retour_client rc

LEFT JOIN ref_magasin rm ON rc.ref_magasin = rm.ref_magasin

WHERE rm.ref_magasin IS NULL;
```

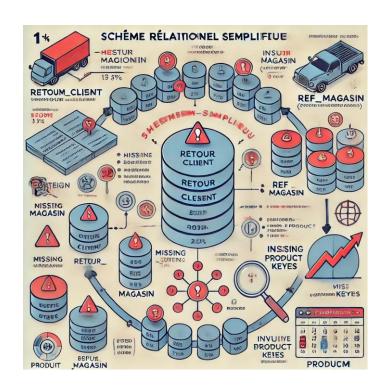
-- Détecter les produits associés à un retour client mais inexistants dans la table produit

SELECT rc.\*

FROM retour client re LEET JOIN produit p ON re cle produit =

FROM retour\_client rc LEFT JOIN produit p ON rc.cle\_produit = p.cle\_produit
WHERE p.cle\_produit IS NULL;

- \* Mise en place d'un Suivi d'Intégrité des Données
  - Automatisation des vérifications avec des contraintes de clés étrangères
  - Définition de règles de validation pour empêcher l'insertion de données incohérentes.
  - Rapports réguliers sur les anomalies détectées pour correction proactive.
  - **Objectif final :** Maintenir des données cohérentes et éviter les erreurs relationnelles impactant les analyses et décisions stratégiques.



# 6) Étapes suivantes

- **Prochaine étape :** Suivi post-déploiement pour évaluer l'efficacité et ajuster les stratégies.
- **Objectif**: Assurer l'adoption efficace des nouveaux outils et optimiser leur usage.

## ♠ Formation des Équipes

### Actions :

- Organisation d'ateliers pratiques sur les tableaux de bord.
- Création de guides et tutoriels pour accompagner les utilisateurs.
- Mise en place d'un support dédié pour répondre aux questions.
- Sensibilisation aux bonnes pratiques d'analyse et de gestion des données.

### Actions :

- Intégration de tableaux de bord interactifs pour suivre la satisfaction client et les tendances des retours.
- Utilisation de Power BI / Tableau / Looker pour une visualisation optimisée.
- Création d'alertes automatisées en cas d'anomalies.
- Personnalisation des rapports selon les besoins des équipes.