

# Retail Insight 360

## 1. Contexte du projet

### Objectif :

L'objectif principal de *RetailInsight360* est d'améliorer la satisfaction client en identifiant les raisons des retours produits et d'analyser les comportements clients (où, quand et comment les retours se produisent).

Ce projet vise également à améliorer la qualité de service et à renforcer la fidélité client.

### Enjeux:

- Identifier les facteurs influençant les retours et le niveau de satisfaction client.
- Optimiser l'expérience client en magasin, en ligne et sur le drive.
- Réduire le taux de retour et améliorer les recommandations (NPS).

### Bénéfices attendus:

- Fidélisation accrue des clients.
- Meilleure gestion des retours produits.
- Amélioration de l'image de BestMarket grâce à des retours positifs sur les réseaux sociaux.

## 2. Expression des besoins

### 2.1. Liste des besoins exprimés par Olivier

Questions posées	Indicateurs associés	Requete MySQL
<b>Question 1:</b> Quel est le nombre de retours clients sur la livraison?	<ul style="list-style-type: none"> <li>• libelle_categorie pour filtrer les retours liés à la livraison.</li> <li>• cle_retour_client (pour compter le nombre de retours). PK</li> </ul>	<pre>SELECT COUNT(cle_retour_client) AS nombre_retours_livraison FROM retour_client WHERE libelle_categorie = 'livraison';</pre>
<b>Question 2:</b> Quelle est la liste des notes des clients sur les réseaux sociaux concernant les télévisions?	<ul style="list-style-type: none"> <li>• cle_produit (FK)</li> <li>• libelle_source (pour identifier "réseaux sociaux").</li> <li>• titre_produit (pour sélectionner "TV").</li> <li>• note (liste des notes des clients).</li> </ul>	<pre>SELECT rc.note AS note_TV_rs 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' GROUP BY rc.note;</pre>
<b>Question 3:</b> Quelle est la note moyenne pour chaque catégorie de produit?	<ul style="list-style-type: none"> <li>• cle_produit (PK)</li> <li>• typologie_produit (regroupement par catégorie).</li> <li>• note (pour calculer la moyenne).</li> </ul>	<pre>SELECT p.typologie_produit, ROUND(AVG(rc.note), 2) AS note_moyenne FROM produit p JOIN retour_client rc ON p.cle_produit = rc.cle_produit GROUP BY p.typologie_produit ORDER BY note_moyenne DESC;</pre>
<b>Question 4:</b> Quels sont les 5 magasins avec les meilleures notes moyennes?	<ul style="list-style-type: none"> <li>• ref_magasin (FK).</li> <li>• libelle_de_commune et département (pour identifier le magasin - données actuellement manquantes).</li> <li>• note (pour classer la moyenne)</li> </ul>	<pre>SELECT rm.libelle_de_commune AS Ville, rm.departement AS dep_code, ROUND(AVG(rc.note),2) AS note_moyenne FROM retour_client rc JOIN ref_magasin rm ON rc.ref_magasin = rm.ref_magasin GROUP BY rc.ref_magasin ORDER BY note_moyenne DESC LIMIT 5;</pre>
<b>Question 5:</b> Quels magasins ont plus de 12 feedbacks sur le drive?	<ul style="list-style-type: none"> <li>• ref_magasin (FK).</li> <li>• libelle_de_commune et département (pour identifier le magasin - données actuellement manquantes).</li> <li>• libelle_source (pour identifier le drive).</li> <li>• recommandation (compte des feedback).</li> </ul>	<pre>SELECT rm.libelle_de_commune AS Ville, rm.departement AS dep, COUNT(rc.recommandation) AS Feedback FROM retour_client rc JOIN ref_magasin rm ON rc.ref_magasin = rm.ref_magasin WHERE LOWER(rc.libelle_categorie) = 'drive' GROUP BY rm.libelle_de_commune, rm.departement HAVING Feedback &gt; 12;</pre>
<b>Question 6:</b> Quel est le classement des départements par note ?	<ul style="list-style-type: none"> <li>• ref_magasin (FK).</li> <li>• département (donnée actuellement manquante).</li> <li>• note.</li> </ul>	<pre>SELECT rm.departement, ROUND(AVG(rc.note), 2) AS note_moyenne FROM retour_client rc JOIN ref_magasin rm ON rc.ref_magasin = rm.ref_magasin GROUP BY departement ORDER BY note_moyenne DESC;</pre>

<p><b>Question 7:</b> Quelle typologie de produit apporte le meilleur service après-vente?</p>	<ul style="list-style-type: none"> <li>• cle_produit (PK)</li> <li>• typologie_produit (depuis la table produit).</li> <li>• note (pour le produit)</li> <li>• recommandation (si malgré le retour le client recommande le service).</li> <li>• libelle_categorie (pour filtrer le service après-vente).</li> </ul>	<pre>SELECT p.typologie_produit, ROUND(AVG(rc.note), 2) AS note_moyenne FROM retour_client rc JOIN produit p ON rc.cle_produit = p.cle_produit WHERE rc.libelle_categorie = 'service après-vente' GROUP BY p.typologie_produit ORDER BY note_moyenne DESC LIMIT 1;  WITH pourcentages AS ( SELECT p.typologie_produit AS categorie, count(rc.cle_retour_client) AS totRetours, -- nombre de retour par typologie / nombre de retours au totale x100 concat(round(count(rc.cle_retour_client) / (SELECT count(*) FROM retour_client WHERE LOWER(libelle_categorie) LIKE 'service%vente') * 100), ' %') AS surTot, count(CASE WHEN rc.recommandation = TRUE THEN 1 ELSE NULL end) AS fdPos, count(CASE WHEN rc.recommandation = FALSE THEN 1 ELSE NULL end) AS fdNeg FROM retour_client rc JOIN produit p ON rc.cle_produit = p.cle_produit WHERE LOWER(rc.libelle_categorie) LIKE 'service%vente' GROUP BY p.typologie_produit) SELECT categorie, -- fdPos * 100 / fdPos + fbNeg concat(round((fdPos*100) / (fdPos + fdNeg)), ' %') AS pourcentagePos FROM pourcentages ORDER BY pourcentagePos DESC LIMIT 1;</pre>
<p><b>Question 8:</b> Quelle est la note moyenne sur l'ensemble des boissons?</p>	<ul style="list-style-type: none"> <li>• cle_produit (FK)</li> <li>• titre_produit (pour filtrer les boissons).</li> <li>• note (pour la moyenne)</li> </ul>	<pre>SELECT ROUND(AVG(rc.note), 2) as note_boissons FROM retour_client rc JOIN produit p ON rc.cle_produit = p.cle_produit WHERE LOWER(p.titre_produit) LIKE 'boisson%';</pre>
<p><b>Question 9:</b> Quel est le classement des jours de la semaine offrant la meilleure expérience client?</p>	<ul style="list-style-type: none"> <li>• date_achat (pour extraire le jour).</li> <li>• note (moyenne par jour).</li> </ul>	<pre>SELECT DAYNAME(date_achat) AS jour, ROUND(AVG(note),2) AS note_moyenne FROM retour_client WHERE LOWER(libelle_categorie) = 'expérience en magasin' GROUP BY jour ORDER BY note_moyenne DESC;</pre>
<p><b>Question 10:</b> Sur quel mois a-t-on le plus de retours après-vente?</p>	<ul style="list-style-type: none"> <li>• date_achat (pour extraire le mois).</li> <li>• cle_retour_client (pour calculer les retours).</li> </ul>	<pre>SELECT MONTH(date_achat) AS mois, COUNT(*) AS nombre_retours FROM retour_client WHERE libelle_categorie = 'service après-vente' GROUP BY mois ORDER BY nombre_retours DESC LIMIT 1;</pre>
<p><b>Question 11:</b> Quel est le pourcentage de recommandations client?</p>	<ul style="list-style-type: none"> <li>• recommandation (valeur binaire ou NULL).</li> <li>• cle_retour_client (total des retours).</li> </ul> <p><b>Calcul:</b> Pourcentage = nombre de recommandations positives / Nombre total de retours x 100</p>	<pre>SELECT CONCAT(ROUND((COUNT(CASE WHEN recommandation = TRUE THEN 1 END) * 100 / COUNT(*)),1), ' %') AS "Positive", CONCAT(ROUND((COUNT(CASE WHEN recommandation = 0 THEN 1 END) * 100 / COUNT(*)),1), ' %') AS "Negative" FROM retour_client;</pre>
<p><b>Question 12:</b> Quels magasins ont une note inférieure à la moyenne?</p>	<ul style="list-style-type: none"> <li>• ref_magasin (FK)</li> <li>• libelle_de_commune et département (pour identifier le magasin - données actuellement manquantes).</li> <li>• note (pour le calcul de la moyenne)</li> </ul>	<pre>SELECT rm.libelle_de_commune, rm.departement, ROUND(AVG(rc.note),2) AS note_magasin FROM retour_client rc JOIN ref_magasin rm ON rc.ref_magasin = rm.ref_magasin GROUP BY rc.ref_magasin HAVING note_magasin &lt; (SELECT AVG(rc.note) FROM retour_client rc) ORDER BY note_magasin ASC;</pre>
<p><b>Question 13:</b> Quelles sont les typologies produits qui ont amélioré leur moyenne entre le 1er et le 2ème trimestre 2021?</p>	<ul style="list-style-type: none"> <li>• cle_produit (FK)</li> <li>• typologie_produit</li> <li>• note (pour la moyenne et le classement)</li> <li>• date_achat (pour filtrer par trimestre).</li> </ul> <p>• <b>taux évolution</b> (pourcentage d'amélioration).</p> <p><b>Calcul:</b> Evolution = (moyenne T2 - moyenne T1) / moyenne T1 x 100</p>	<pre>WITH medie AS ( SELECT p.typologie_produit AS categorie, AVG(CASE WHEN MONTH(rc.date_achat) BETWEEN 1 AND 3 THEN rc.note ELSE NULL END) AS moyenneT1, AVG(CASE WHEN MONTH(rc.date_achat) BETWEEN 4 AND 6 THEN rc.note ELSE NULL END) AS moyenneT2 FROM produit p JOIN retour_client rc ON p.cle_produit = rc.cle_produit GROUP BY p.typologie_produit) SELECT categorie, ROUND(moyenneT1, 2) AS moyenneT1, ROUND(moyenneT2, 2) AS moyenneT2, CONCAT(ROUND(((moyenneT2 - moyenneT1) / moyenneT1 *100), 2), ' %') AS taux FROM medie WHERE moyenneT2 &gt; moyenneT1;</pre>
<p><b>Question 14:</b> Quel est le NPS global?</p>	<ul style="list-style-type: none"> <li>• note (pour classer les retours: promoteurs et detracteurs).</li> <li>• NPS: %promoteurs - %detracteurs</li> </ul> <p><b>Calcul:</b> NPS = %promoteurs - %detracteurs avec promoteurs: note&gt;=9 et detracteurs: note &lt;=6</p>	<pre>SELECT ROUND((promoteurs * 100.0 / tot) - (detracteurs * 100.0 / tot), 2) AS NPS_global FROM (SELECT SUM(CASE WHEN note &gt; 8 THEN 1 ELSE 0 END) AS promoteurs, SUM(CASE WHEN note &lt;= 6 THEN 1 ELSE 0 END) AS detracteurs, COUNT(note) AS tot FROM retour_client);</pre>
<p><b>Question 15:</b> Quel est le NPS par source (réseaux sociaux, téléphone, etc.)?</p>	<ul style="list-style-type: none"> <li>• libelle_source.</li> <li>• note.</li> </ul> <p><b>Calcul:</b> NPS = %promoteurs - %detracteurs avec promoteurs: note&gt;=9 et detracteurs: note &lt;=6</p>	<pre>SELECT source, ROUND((promoteurs * 100.0 / tot) - (detracteurs * 100.0 / tot)) AS NPS_par_source FROM (SELECT SUM(CASE WHEN note &gt; 8 THEN 1 ELSE 0 END) AS promoteurs, SUM(CASE WHEN note &lt;= 6 THEN 1 ELSE 0 END) AS detracteurs, COUNT(note) AS tot, libelle_source AS source FROM retour_client GROUP BY libelle_source) AS stats;</pre>
<p><b>Question 16:</b> Quel est le nombre de retours clients par source?</p>	<ul style="list-style-type: none"> <li>• libelle_source.</li> <li>• cle_retour_client.</li> </ul>	<pre>SELECT libelle_source, COUNT(*) AS nombre_retours FROM retour_client GROUP BY libelle_source;</pre>

<p><b>Question 17:</b> Quels sont les 5 magasins avec les plus de feedbacks?</p>	<ul style="list-style-type: none"> <li>• ref_magasin (FK).</li> <li>• libelle_de_commune (pour identifier le magasin - donnée actuellement manquante).</li> <li>• note (pour classer la moyenne)</li> </ul>	<pre>SELECT rm.libelle_de_commune, rc.ref_magasin, COUNT(rc.recommandation) AS reviews FROM retour_client rc JOIN ref_magasin rm ON rc.ref_magasin = rm.ref_magasin GROUP BY rm.libelle_de_commune, rc.ref_magasin ORDER BY reviews DESC LIMIT 5;</pre>
<p><b>Question 18:</b> Quelle est la différence entre les feedbacks reçus sur le drive et en magasin?</p>	<ul style="list-style-type: none"> <li>• libelle_categorie (pour distinguer le drive et le magasin).</li> <li>• recommandation.</li> </ul>	<pre>SELECT libelle_categorie, COUNT(CASE WHEN recommandation = TRUE THEN 1 END) AS fb_positive FROM retour_client WHERE LOWER(libelle_categorie) = 'drive' OR libelle_categorie LIKE '%magasin' GROUP BY libelle_categorie;</pre>
<p><b>Question 19:</b> Quel est le classement des notes pour chaque catégorie?</p>	<ul style="list-style-type: none"> <li>• cle_produit (PK)</li> <li>• typologie_produit</li> <li>• titre_produit (depuis la table produit).</li> <li>• note (table retour_client)</li> </ul>	<pre>WITH Classement AS ( SELECT p.typologie_produit AS TypologieProduit, p.titre_produit AS Produit, ROUND(AVG(rc.note),2) AS NoteMoyenne, ROW_NUMBER() OVER ( PARTITION BY p.typologie_produit ORDER BY AVG(rc.note) DESC) AS Classement FROM produit p JOIN retour_client rc ON p.cle_produit = rc.cle_produit GROUP BY p.typologie_produit, p.titre_produit) SELECT TypologieProduit, Produit, NoteMoyenne, Classement FROM Classement WHERE Classement &lt;= 5 ORDER BY TypologieProduit, Classement;</pre>
<p><b>Question 20:</b> Quel est le pourcentage de retour pour chaque typologie de produit?</p>	<ul style="list-style-type: none"> <li>• typologie_produit.</li> <li>• cle_retour_client (pour calculer les retours).</li> </ul> <p><b>Calcul:</b> Pourcentage =  nombre de retours typologie / nombre de retours x 100</p>	<pre>SELECT p.typologie_produit AS Produit, COUNT(rc.cle_retour_client) AS Retours, CONCAT(ROUND(COUNT(rc.cle_retour_client) / (SELECT COUNT(*) FROM retour_client) * 100), "%") AS "%retours" FROM produit p JOIN retour_client rc ON p.cle_produit = rc.cle_produit GROUP BY p.typologie_produit ORDER BY "%retours" DESC;</pre>
<p><b>Question 21:</b> Quel est le nombre de retours négatifs par ville, rapporté à la population de chaque ville?</p>	<ul style="list-style-type: none"> <li>• libelle_de_commune (nom de la ville - donnée actuellement manquante).</li> <li>• population (tot habitants de la ville - donnée actuellement manquante).</li> <li>• recommandation (pour retour négatifs FALSE)</li> </ul> <p><b><u>Proportion de retours négatifs par rapport la population.</u></b></p> <p><b>Calcul:</b> Proportion =  nombre de retours négatifs / population de la ville x 100</p>	<pre>SELECT rm.libelle_de_commune AS Ville, rm.population AS Population_totale, COUNT(CASE WHEN rc.recommandation = FALSE THEN 1 END) AS Retours_negatifs, ROUND((COUNT(CASE WHEN rc.recommandation = FALSE THEN 1 END) / rm.population) * 100, 2) AS Proportion FROM retour_client rc JOIN ref_magasin rm ON rc.ref_magasin = rm.ref_magasin GROUP BY rm.libelle_de_commune, rm.population HAVING Retours_negatifs &gt; 0 ORDER BY Proportion DESC;</pre>

## 2.2. Suggestions d'indicateurs supplémentaires

Mes suggestions supplémentaires sont marquées en rouge: **questions 18, 19, 20 et 21.**

Mes indicateurs supplémentaires sont les suivants:

1. **Proportion de retours négatifs:** pourcentage de retours négatifs par rapport à la population totale de la ville.
2. **Taux d'évolution:** pourcentage d'amélioration des notes de produits sur une base trimestrielle.
3. **Taux de retour:** la pourcentage de retours par typologie sur le total des produits vendus dans cette typologie (si l'on disposait du total des produits vendus).

En jaune sont marqués les questions qui auraient nécessité de plus des informations, concernant les magasins, (comme des données géographiques ou la population).

## 3. Exigences du projet

### 3.1. Exigences techniques

- **Ajout des adresses et des données des magasins.** Actuellement, seules les références des magasins (IDs) sont disponibles dans la base de données. Disposer d'informations géographiques et régionales détaillées permet d'analyser les performances des magasins en fonction de leur localisation et de la population desservie, tout en facilitant la création de rapports clairs et pertinents pour des comparaisons géographiques approfondies.
- **Ajouter des données sur tous les produits vendus.** Pour permettre un calcul précis du taux de retour par produit, il est nécessaire d'intégrer des données détaillées sur tous les produits vendus. Cela inclut les quantités totales de produits écoulés, afin de comparer efficacement le nombre de retours clients par rapport au volume total des ventes pour chaque produit. Cette information est essentielle pour identifier les catégories ou produits présentant un taux de retour élevé et optimiser les performances globales.

### 3.2. Exigences temporelles

- Livraison du rapport final pour la phase initiale dans un délai d'1 semaine.

## Conclusion:

Le projet *RetailInsight360* représente une opportunité majeure pour *BestMarket* d'améliorer ses performances en termes de service client et d'analyser précisément les retours. Le respect des exigences techniques, humaines et temporelles garantira le succès du projet.