Giulia Governatori pour BestMarket

Olivier: Responsable Service Client

Date: 24/01/25

Rapport de Cohérence des Données: Retail Insight 360

1. Vérification de l'unicité des clés primaires

Toutes les clés primaires des tables `retour_client`, `produit` et `ref_magasin` sont uniques. Aucun doublon n'a été détecté.

Unicité des clés primaires de la table `retour_client` :

```sql
SELECT cle_retour_client, COUNT(*) AS total
FROM retour_client
GROUP BY retour_client
HAVING total > 1;
```

Résultat : Aucun enregistrement trouvé, les clés primaires sont uniques.

Unicité des clés primaires de la table `produit` :

```sql
SELECT cle_produit, COUNT(*) AS total
FROM produit
GROUP BY cle_produit
HAVING total > 1;
```

Résultat : Aucun enregistrement trouvé, les clés primaires sont uniques.

Unicité des clés primaires de la table `ref_magasin` :

```sql
SELECT ref_magasin, COUNT(*) AS total
FROM ref_magasin
GROUP BY ref_magasin
HAVING total > 1;
```

Résultat : Aucun enregistrement trouvé, les clés primaires sont uniques.

2. Vérification des clés étrangères orphelines

Toutes les clés étrangères (`cle_produit` et `ref_magasin`) de la table `retour_client` ont une correspondance valide dans leurs tables respectives. Aucun enregistrement orphelin n'a été trouvé.

Validation des correspondances entre `retour_client` et `produit`:

```sql
SELECT rc.cle_produit
FROM retour_client rc
RIGHT JOIN produit p ON rc.cle_produit = p.cle_produit
WHERE p.cle_produit IS NULL;

SELECT rc.cle_produit
FROM retour_client rc
LEFT JOIN produit p ON rc.cle_produit = p.cle_produit
WHERE rc.cle_produit IS NULL;
```

Validation des correspondances entre `retour_client` et `ref_magasin`:

```sql
SELECT rc.ref_magasin
FROM retour_client rc
LEFT JOIN ref_magasin rm ON rc.ref_magasin = rm.ref_magasin
WHERE rc.ref_magasin IS NULL;

SELECT rc.ref_magasin
FROM retour_client rc
RIGHT JOIN ref_magasin rm ON rc.ref_magasin = rm.ref_magasin
WHERE rm.ref_magasin IS NULL;
```

Résultat : Aucun enregistrement orphelin trouvé.

3. Vérification des valeurs anormales

Les valeurs des champs ont été contrôlées pour rester dans les plages attendues (exemple : les notes entre 0 et 10). Aucun enregistrement anormal n'a été détecté.

```sql
SELECT COUNT(*) AS combienNote_0
FROM retour_client
WHERE note < 0 OR note > 10;
```

Résultat : Aucun enregistrement anormal trouvé, toutes les notes sont conformes.

---

4. Vérification des doublons

Aucune duplication de données n'a été trouvée dans les enregistrements (`retour_client` et `ref_magasin`) ni dans les coordonnées GPS des magasins.

```sql
SELECT cle_retour_client, cle_produit, date_achat,
COUNT(*) AS total
FROM retour_client
GROUP BY cle_retour_client, cle_produit, date_achat
HAVING total > 1;
```

Contrôle des doublons dans les coordonnées GPS des magasins.

```sql
SELECT geo_point_2d,
COUNT(*) AS total
FROM ref_magasin
GROUP BY geo_point_2d
HAVING total > 1;
```

Résultat : Aucun doublon détecté.

---

5. Contrôle des valeurs attendues dans les champs

Tous les valeurs des champs ont été vérifiés.

```sql
SELECT libelle_source
FROM retour_client
GROUP BY libelle_source;
```

```sql
SELECT recommandation, COUNT(*)
FROM retour_client
GROUP BY recommandation
ORDER BY recommandation;
```

Résultat: `recommandation` contient 674 valeurs vides, 211 valeurs `FALSE` et 2 115 valeurs `TRUE`. Toutes les autres colonnes respectent les valeurs attendues.

---

6. Vérification de la distribution des données

Tous les produits ont une typologie valide. Chaque magasin a un département, une population et une commune associée. Aucun déséquilibre ou absence de données n'a été détecté.

Exemple: chaque produit appartient à une catégorie

```sql
SELECT cle_produit, typologie_produit
FROM produit
WHERE typologie_produit IS NULL
OR typologie_produit = '';
```

Résultat : Aucun enregistrement manquant, tous les produits ont une catégorie valide.

En outre:

- Chaque magasin a un département, une population et une commune.
- Chaque commune compte un seul magasin.
- Chaque produit a un titre et un typologie.
- Chaque retour a son identifiant produit, son identifiant magasin, sa "date_achat", "libelle_source", et "libelle_categories".

7. Cohérence entre qualité produit, note et recommandation:

Les données de `note`, `recommandation` et catégorie `qualité produit` ont été croisées pour vérifier leur cohérence.

```
SELECT
      COUNT(*) AS Retours,
    CASE
        WHEN (note < 7 AND recommandation = TRUE)
        THEN 'Incohérence: note basse mais recommandation positive'
        WHEN (note >= 7 AND recommandation = FALSE)
        THEN 'Incohérence: note élevée mais recommandation négative'
        WHEN libelle_categorie = 'qualité produit'
        AND (note < 7 AND recommandation = FALSE)
        THEN 'Incohérence: note basse, recommandation négative mais marqué de
bonne qualité'
        ELSE 'Cohérent'
    END AS coherence_status
FROM retour_client
GROUP BY coherence_status
ORDER BY coherence_status;
```

Résultat:

2150	Cohérent
105	Incohérence: note basse mais recommandation positive
26	Incohérence: note basse, recommandation négative mais marqué de bonne qualité
719	Incohérence: note élevée mais recommandation négative

8. L'ajout des contraints à les tables:

Afin de garantir l'unicité des clés à l'avenir, et donc d'éviter d'éventuelles erreurs, nous ajoutons différentes contraintes.

Unicité des clés primaires (PK):

```
ALTER TABLE retour_client ADD UNIQUE (cle_retour_client);
```

Unicité de relation avec les clés étrangères (FK):

`retour_client + produit:`

```
ALTER TABLE retour_client ADD CONSTRAINT fk_cle_produit
FOREIGN KEY (cle_produit) REFERENCES produit(cle_produit);

ALTER TABLE retour_client ADD UNIQUE (cle_retour_client, cle_produit);
```

`retour_client + ref_magasin:`

```
ALTER TABLE retour_client ADD CONSTRAINT fk_ref_magasin
FOREIGN KEY (ref_magasin) REFERENCES ref_magasin(ref_magasin);

ALTER TABLE retour_client ADD UNIQUE (cle_retour_client, ref_magasin);
```