



AXE 1 : Gestion des clients fidélité de la supérette

Les liens :

« Client » et « Ville » : 1 client vit dans une ville mais une ville peut avoir plusieurs clients qui le réside.

« Client » et « CarteFidelite » : 1 client peut d'avoir une carte de fidélité, mais une carte de fidélité appartient forcément à un client, pour avoir la carte il faut les données du client, sa ville par exemple (contrainte d'inclusion).

« CarteFidelite » et « Ticket » : une carte de fidélité peut avoir des achats et les achats pourraient être enregistrer dans la carte de fidélité (si le client fait son achat avec sa carte de fidélité).

« Ticket » et « Caisse » : les achats sont réglés à la caisse, une caisse peut avoir plusieurs achats qui sont réglés.

« Ticket » et « Produit » : les achats sont les produits achetés.

« Produit » et « Rayon » : un produit appartient forcément à un rayon mais un rayon peut avoir plusieurs produits.

« CarteFidelite » et « StatutFidel » : une carte a son statut de fidélité définit par ses points de fidélités qui les a cumulés et un statut appartient ou pas aux plusieurs clients.

« Client » et « Evènement » : un client peut participer à un évènement seulement s'il possède une carte de fidélité et un statut de fidélité (client fidèle) afin de participer aux évènements (contrainte d'inclusion), un évènement peut avoir plusieurs clients fidèles.

« Evènement » et « BonPromotionnel » : un évènement propose forcément des bon de promotionnel, et un bon promotionnel peut être utilisé pendant l'évènement seulement.

« BonPromotionnel » et « Produit » : ces bons promotionnels vont réduire sur le prix des produits selon le stock limité.

Proposition : motivé les clients fidèles à faire plus d'achat en mettant un système qui va classer son rang selon son point de fidélité comparer à d'autres clients fidèles, cela créer une ambiance compétitive entre clients qui cherchent être le meilleur.

```
Select c.Code_Cli, c.Nom, c.Prenom, cf.PointsFidelite, RANK() OVER (ORDER BY cf.PointsFidelite
DESC) AS Rang_Fidelite
```

```
FROM Client c
```

```
JOIN CarteFidelite cf ON c.Code_Cli = cf.Code_Cli
```

```
ORDER BY Rang_Fidelite;
```

AXE 2: Gestion de dépôts de commandes

« Client » et « Commande » : un client peut commander une ou plusieurs commande mais une commande appartient seulement à un client.

« Commande » et « PointDepot » : une commande est livrée vers un point dépôt, un point de dépôt peut avoir plusieurs commandes qui sont livrées, une commande peut être livrée seulement si elle a une heure de passage et le client avait déjà réglé son achat à la caisse.

« Commande » et « Tournee » : une commande a son horaire de passage, une tournée peut avoir plusieurs commandes qui livrent vers un même point de dépôt « PointDepot ».

« PointDepot » et « horaires arrivés », un point de dépôt enregistre tous les livraisons qui sont arrivés.

« horaires arrivés » et « tournée » vers « Client » : un client est informé, le temps de départ de sa commande et son temps arrivé estimé, le temps arrivé de la commande ,la commande pourrait ou ne pas arriver qui pourrait dû au accident ou commande défectueux.

Proposition : voir quels jours de la semaine il y a plus de commandes afin d'organiser mieux la tournée afin que la livraison arrive au plutôt possible.

```
SELECT Date_Tournee, COUNT(*) AS nb_tournees
FROM Tournee
GROUP BY Date_Tournee;
```

AXE 3 : Ateliers dégustation

« Client » et « Atelier » : un client peut s'inscrire à un ou plusieurs ateliers, en même temps un atelier peut accueillir plusieurs clients.

« Atelier » et « bonReduction » : après l'atelier les bons de réduction peuvent être proposer, un bon de réduction peut appartenir à un atelier

Proposition : on veut savoir quel thème a le plus d'inscription afin de modérer les bons de réduction.

```
SELECT a.Thème, COUNT(i.Code_Cli) AS Nb_Inscrits
FROM Atelier a
LEFT JOIN Inscription i ON a.Code_Atelier = i.Code_Atelier
GROUP BY a.Thème
ORDER BY Nb_Inscrits DESC;
```