Eya LAHOUEL - Mohamed Ilyes BEN SAID – Mohamed SAKHO

Rapport de Conception de Base de Données E-Commerce

Introduction et Problématique Client

1. Contexte du Projet

Ce rapport synthétise la première phase de conception d'un système d'information destiné à gérer une activité de vente au détail en ligne (e-commerce). L'objectif est de mettre en place une **Base de Données Relationnelle** robuste et normalisée capable de soutenir l'ensemble des processus énoncé par l'entreprise de commerce.

2. Énoncé du Problème

La complexité du modèle économique de l'entreprise nécessite un système capable de gérer les contraintes suivantes :

- **Gestion des variations de produits (SKU) :** Les produits existent en multiples variations (taille, couleur) qui doivent être traitées comme des unités de stock et de vente distinctes.
- **Historisation des données :** Nécessité de tracer l'historique des prix (promotions) et de capturer le prix exact payé par le client lors de la commande (immutabilité financière).
- **Logistique et Inventaire :** Le stock doit être géré par variation d'article et par lieu de stockage (entrepôt ou magasin physique).
- **Relation Client :** Gestion multi-adresses, préférences de communication, et un cycle complet de support après-vente (commandes, retours, tickets de support, avis).

Démarche de Modélisation

1. Modèle Conceptuel des Données (MCD)

Le MCD a permis d'identifier l'ensemble des entités et des relations entre elles. Les choix principaux au niveau conceptuel ont été :

- **Séparation Produit/Article :** Distinction entre l'entité **PRODUCT** (le modèle conceptuel : nom, description, marque) et l'entité **ARTICLE_SKU** (la variation physique : taille, couleur, référence de stock).
- **Associative Entities (Entités-Associations) :** La gestion des relations de type =N:N a conduit à la création d'entités-associations pour plus de clarté et pour porter des attributs spécifiques.

2. Modèle Logique des Données (MLD)

La conversion du MCD en MLD (fait par looping but we could have made it) a transformé les relations en **Clés Étrangères (FK)**. Les choix critiques qui garantissent l'intégrité et la fonctionnalité du modèle sont détaillés ci-dessous, y compris une explication des réflexions qui nous ont menées à ces choix.

2.1. Distinction Produit (PRODUCT) vs Article (ARTICLE_SKU)

Entité	Rôle	Justification
PRODUCT	Le concept, le modèle	Sert au **marketing** (description, marque, catégorie) et porte les **avis** (REVIEW).
ARTICLE_SKU	La variation unique	Sert aux **opérations** et aux **ventes** (prix, stock, commande).

Justification : Cette distinction est la réponse directe au besoin de lier le contenu marketing (décrire le produit) aux données logistiques (stocker et vendre la variation). Sans cette séparation, des informations textuelles volumineuses seraient dupliquées sur chaque ligne de variation d'article. Les produits ne sont pas considéré de manière totalement atomique mais en groupe d'une même taille et couleur.

2.2. Réflexion et Normalisation sur les Adresses (3e Forme Normale)

Problématique initiale : Dans l'ébauche du modèle, l'entité WAREHOUSE possédait un attribut address (adresse complète).

- **Réflexion :** Si l'on décompose l'adresse de l'entrepôt (Address_Line, City, Postal_Code), le modèle risque de violer la **1FN et donc la 3FN**. Un attribut comme City dépendrait de Postal_Code et non directement de l'identifiant de l'entrepôt. De plus, il y avait une duplication de la structure d'adresse entre l'entité ADDRESS (pour le client) et l'entité WAREHOUSE.
- **Solution adoptée :** Nous avons décidé de créer une entité générique et isolée nommée **LOCATION** pour stocker tous les détails géographiques (Address_Line, City, Postal_Code, Country).
    - La table WAREHOUSE référence désormais cette entité via la clé étrangère **#Location_ID**.
    - L'ancienne table ADDRESS pour le client est renommée **CLIENT_ADDRESS** et référence également **#Location_ID**.

Cette refonte garantit la **3FN** et centralise la gestion des données géographiques (pas de duplication des villes et codes postaux).

2.3. Gestion du Stock et Immuabilité

Problématique de la commande : Lors d'un retour client, le système doit garantir que la quantité d'articles retournés est logique et que les références sont exactes.

- **Réflexion :** La règle métier stipule que l'on achète une **quantité** d'un type d'article (SKU). Lors d'un retour, même si l'attribut SKU_Code identifie le type de produit, la **quantité retournée** doit être enregistrée pour mettre à jour correctement les stocks et calculer le remboursement.
- **Solution adoptée :** L'entité associative **RETURNED_ITEM** utilise la quantité (quantity_returned) pour enregistrer le nombre d'unités d'une même référence

(#SKU_code) retournées dans le cadre d'un seul événement de retour (#return_id). Cela évite de créer des lignes redondantes pour chaque unité retournée.

2.4. Le Cycle Commande - Ligne - Transaction

- **ORDER_LINE (Détail de la Commande) :** L'attribut **unit_price_paid** est essentiel. Il enregistre le prix de l'article *à l'instant de la vente*.
    - **Justification :** Si le prix catalogue change après la commande (exemple : en fin de promotion), ce champ garantit l'intégrité de la transaction et la justesse du montant en cas de litige ou de retour.
- **Séparation ORDER vs TRANSACTION :** Sépare l'acte commercial de la preuve financière, permettant de gérer les statuts d'une commande même en cas de paiement différé ou échoué.

2.5. Le Cycle Support Client

Les entités **SUPPORT_TICKET** et **TICKET_HISTORY** fonctionnent comme un système d'audit :

- **SUPPORT_TICKET** : Le **dossier principal**, contenant le statut *actuel* et le sujet.
- **TICKET_HISTORY** : Le **journal d'audit**, enregistrant chaque échange, chaque changement de statut avec un la date et l'heure précise.

Justification : Ce modèle respecte les exigences énoncées.

Identification des Entités Fortes et Faibles

Le modèle présente des cas clairs d'entités fortes et faibles, définis par des relations d'identification (où la clé primaire de l'entité faible est composée de sa propre clé et de la clé de l'entité forte).

Entité Faible	Entité Forte	Clé Primaire de l'Entité Faible	Dépendance
TICKET_HISTORY	**SUPPORT_TICKET**	#Ticket_id, line_id	Une ligne d'historique n'a de sens que dans le contexte d'un ticket spécifique.
PRICE	**ARTICLE_SKU**	#sku_code, start_date	Un prix est identifié par l'article et la date à laquelle il entre en vigueur.

Justification : Ce choix modélise la dépendance structurelle et assure que les données conversation et prix historique ne peuvent exister sans leur entité parente (le ticket, l'article), renforçant ainsi l'intégrité référentielle.

« Conclusion »

Le Modèle Logique de Données est le résultat d'une démarche de normalisation rigoureuse. L'intégration de l'entité **LOCATION** pour garantir la 3FN sur les adresses, combinée une

structure hiérarchique du produit et une gestions des actions financières fournit une base de données performante, flexible et prête à supporter la croissance de l'entreprise.