

## 5.2 Base de données (vue logique)

Cette section présente le modèle logique de données du système GestionMagasin. Elle décrit les principales entités persistées, leurs attributs essentiels et les relations qui les lient. L'objectif est de fournir une vue structurée et indépendante de tout SGBD spécifique, tout en restant fidèle à l'implantation MySQL du projet.

### 5.2.1 Entités principales

#### app\_user

Cette entité représente les comptes utilisateurs de l'application (clients, managers, administrateurs).

Attributs principaux :

id : identifiant unique de l'utilisateur (PK) ;

email : identifiant de connexion unique ;

password\_hash : mot de passe haché (BCrypt) ;

full\_name : nom complet ;

is\_active : indicateur d'activation du compte ;

delivery\_address : adresse de livraison par défaut ;

created\_at : horodatage de création.

#### app\_role

Cette entité modélise les rôles fonctionnels disponibles dans le système (par ex. ADMIN, MANAGER, CLIENT).

Attributs principaux :

id : identifiant unique du rôle (PK) ;

code : code fonctionnel du rôle ;

label : libellé descriptif ;

is\_system : indicateur de rôle système (non supprimable).

#### app\_user\_role

Entité d'association assurant la relation N–N entre app\_user et app\_role.

Attributs principaux :

user\_id : référence à app\_user (FK) ;

role\_id : référence à app\_role (FK).

#### catalog\_product

Cette entité représente les produits commercialisés par le magasin.

Attributs principaux :

id : identifiant unique du produit (PK) ;

sku : référence interne ;

name : nom du produit ;

category\_txt : catégorie textuelle ;

price : prix de vente unitaire ;

stock\_qty : quantité en stock ;

is\_active : indicateur de disponibilité commerciale ;

image\_url : URL de l'image illustrant le produit.

#### **promotion**

Cette entité modélise les codes promotionnels applicables aux paniers ou aux ventes.

Attributs principaux :

id : identifiant unique (PK) ;

code : chaîne de caractères unique identifiant la promotion ;

description : description fonctionnelle ;

discount\_type : type de remise (pourcentage ou montant fixe) ;

discount\_value : valeur de la remise ;

min\_purchase\_amount : montant minimal d'achat pour être éligible ;

start\_date, end\_date : période de validité ;

max\_uses, current\_uses : plafond et compteur d'utilisation ;

is\_active : indicateur d'activation de la promotion ;

created\_at : horodatage de création.

#### **party\_customer**

Entité représentant les clients (personnes ou organisations) associés aux ventes.

Attributs principaux :

id : identifiant unique (PK) ;

name : nom du client ;

phone : numéro de téléphone ;

email : courriel ;

created\_at : date de création du dossier client.

#### **party\_supplier**

Entité représentant les fournisseurs auprès desquels le magasin se réapprovisionne.

Attributs principaux :

id : identifiant unique (PK) ;

name : nom du fournisseur ;

phone : numéro de téléphone ;

email : courriel ;

created\_at : date de création du dossier fournisseur.

#### **sell\_sale**

Cette entité correspond à une vente (commande validée).

Attributs principaux :

id : identifiant unique de la vente (PK) ;

customer\_id : référence au client (party\_customer, FK) ;

sale\_date : date de la vente ;

total\_amount : montant total facturé ;

status : statut de la vente (ex. DRAFT, PAID) ;

created\_by : référence à l'utilisateur ayant enregistré la vente (app\_user, FK) ;

delivery\_address : adresse de livraison associée à la commande ;

created\_at : horodatage de création.

#### **sell\_sale\_line**

Entité représentant les lignes d'articles d'une vente.

Attributs principaux :

id : identifiant unique (PK) ;

sale\_id : référence à la vente (sell\_sale, FK) ;

product\_id : référence au produit (catalog\_product, FK) ;

qty : quantité vendue ;

unit\_price : prix unitaire appliqué.

#### **purchase**

Cette entité modélise les opérations d'achat auprès des fournisseurs (réception de stock).

Attributs principaux :

id : identifiant unique (PK) ;

product\_id : référence au produit (catalog\_product, FK) ;

supplier\_id : référence au fournisseur (party\_supplier, FK) ;

qty : quantité achetée ;

unit\_price : prix d'achat unitaire ;

created\_by : utilisateur ayant enregistré l'achat (app\_user, FK) ;

created\_at : horodatage de création.

stock\_movement

Entité centralisant l'historique des mouvements de stock.

Attributs principaux :

id : identifiant unique (PK) ;

product\_id : référence au produit (catalog\_product, FK) ;

movement\_type : type de mouvement (IN pour entrée, OUT pour sortie) ;

qty : quantité mouvementée ;

reason : motif (par ex. PURCHASE, SALE) ;

ref\_type : type de document d'origine (par ex. "purchase", "sale") ;

ref\_id : identifiant du document d'origine ;

note : commentaire libre ;

created\_by : utilisateur ayant enregistré le mouvement (app\_user, FK) ;

created\_at : horodatage de création.

ai\_config

Entité de configuration de l'assistant IA, sous forme de paires clé/valeur.

Attributs principaux :

config\_key : clé de configuration (PK) ;

config\_value : valeur associée ;

updated\_at : date de dernière mise à jour.

## 5.2.2 Relations et vue d'ensemble

D'un point de vue relationnel, le modèle s'organise autour des interactions suivantes :

Gestion des utilisateurs et des rôles

Relation 1–N entre app\_user et app\_user\_role ;

Relation 1–N entre app\_role et app\_user\_role ;

Ce schéma permet d'associer un nombre quelconque de rôles à un même utilisateur.

Gestion du catalogue et des promotions

`catalog_product` constitue l'entité centrale pour les opérations commerciales.

`promotion` n'est pas directement reliée physiquement aux ventes, mais appliquée logiquement sur les montants de panier/commande.

Gestion des clients, fournisseurs, ventes et achats

Une relation 1–N relie `party_customer` à `sell_sale` (un client peut avoir plusieurs ventes).

Une relation 1–N relie `party_supplier` à `purchase` (un fournisseur peut être lié à plusieurs achats).

Une relation 1–N relie `sell_sale` à `sell_sale_line` (une vente comporte plusieurs lignes).

Chaque `sell_sale_line` référence un `catalog_product`, assurant la traçabilité des articles vendus.

Gestion du stock

`purchase` génère des mouvements de type IN dans `stock_movement` pour le produit concerné.

`sell_sale` (et ses lignes) génère des mouvements de type OUT.

La table `stock_movement`, reliée à `catalog_product`, fournit une vue auditée et chronologique de toutes les entrées et sorties de stock.

Traçabilité des actions

Plusieurs entités (`sell_sale`, `purchase`, `stock_movement`) référencent `app_user` via l'attribut `created_by`, ce qui permet d'identifier l'utilisateur à l'origine de l'opération.

Ainsi, le schéma logique de la base de données assure :

l'intégrité référentielle entre les différentes entités du domaine (produits, ventes, achats, stock, acteurs) ;

la traçabilité complète des mouvements de stock ;

la séparation claire entre gestion des accès (utilisateurs/rôles) et gestion métier (catalogue, ventes, achats, promotions).