

BASE DE DONNÉES

Rapport de Saé

Partie 1

RAPPORT DE

**DA VEIGA SANCHES
CARVALHO FLAVIO**

AUXILIEN KATIA

An abstract graphic in the bottom right corner consisting of numerous thin, dark grey lines that curve and overlap, creating a sense of depth and movement, resembling a stylized wave or a series of concentric arcs.

Hypothèses

Hypothèses sur des informations non précisées dans l'énoncé.

On suppose que chaque personne possède un compte qui peut servir pour la vente et l'achat.

On suppose que le site LivreEchange.com propose les modes de livraison : normale, recommandée, relais, relais express.

On suppose que les frais de livraison sont calculés en fonction du mode de livraison du vendeur.

Compte

Super-classe

Possède des informations d'une personne qui est inscrite sur LivreEchange.com .

Attributs

numCompte : INT NOT NULL

nom : VARCHAR NOT NULL

adressePostale : VARCHAR NOT NULL

numeroTelephone : INT NOT NULL

adresseElectronique : VARCHAR NOT NULL

Contraintes sur les attributs

Tous les attributs de la classe **Compte** ont la condition "NOT NULL" car pour ouvrir un compte sur LivreEchange.com ces attributs sont nécessaires. numCompte est unique et obligatoire parce que c'est l'identifiant de la classe.

Placement des attributs

Ces attributs sont placés dans la super-classe **Compte**, car ils sont hérités par les sous-classes **Vendeur** et **Acheteur**.

Identifiant de la classe

On identifie chaque compte par **numCompte**.

Choix d'une super-classe

Une super-classe et de sous-classes sont adaptés à la création d'un compte sur Livreexchange.com, car il peut y avoir deux types de compte différents, ou les deux en même temps.

Héritage

La super-classe **Compte** à pour héritage une *totalité*, car soit le compte sera acheteur, soit vendeur, soit les deux. Une personne peut acheter et vendre des livres sur le site.

Vendeur

Sous-classe de Compte/ Classe

Possède les informations d'un compte vendeur.

Attributs

numVendeur : INT NOT NULL

modeLivraison : {normale, recommandée, relais, relais express} NOT NULL

Attributs issus de l'héritage de Compte :

numCompte : INT NOT NULL

nom : VARCHAR NOT NULL

adressePostale : VARCHAR NOT NULL

numeroTelephone : INT NOT NULL

adresseElectronique : VARCHAR NOT NULL

Contraintes sur les attributs

numVendeur ne doit pas être vide et il doit être unique, car il s'agit de l'identifiant de la classe Vendeur.

modeDeLivraison ne doit pas être nul, parce qu'on suppose que lors de l'achat, le site doit avoir la donnée sur le mode de livraison du vendeur, ce qui est nécessaire pour l'estimation de la date de livraison, et des frais de livraison que l'acheteur devra régler en même temps que le prix du livre. De plus, l'acheteur a 4 choix possibles, livraison normale, recommandée, relais ou relais express. *Les attributs issus de l'héritage sont déjà remplis puisque les contraintes précédentes ont été appliquées.*

Placement des attributs

Ce sont des informations concernant chaque vendeur, deux vendeurs n'auront pas forcément le même mode de livraison. De plus, il faut pouvoir identifier et avoir les coordonnées de chaque vendeur, par exemple pour envoyer un mail de validation lorsqu'un acheteur déclenchera la procédure d'achat, d'où l'héritage avec la classe association **Compte**. De ce fait, à partir du mode de livraison et de l'adresse postale du vendeur, nous pouvons calculer les frais de livraison à partir de la distance entre le vendeur et l'acheteur et effectuer d'autres procédures à partir de ces données.

Identifiant de la classe

numVendeur est un identifiant de la classe vendeur, car lorsqu'un compte vendeur est créé, un numéro unique lui est attribué.

CompteAcheteur

Sous-classe de Compte / Super-classe

Possède les informations d'un compte acheteur.

Attributs

numAcheteur INT NOT NULL

carteCredit VARCHAR NOT NULL

nomBanque VARCHAR NOT NULL

Attributs issus de l'héritage de la classe Compte :

numCompte : INT NOT NULL

nom : VARCHAR NOT NULL

adressePostale : VARCHAR NOT NULL

numeroTelephone : INT NOT NULL

adresseElectronique : VARCHAR NOT NULL

Contraintes sur les attributs

Les attributs de la classe sont tous requis, ce sont des éléments importants pour la procédure d'achat. **numAcheteur** est l'identifiant du compte, **carteCredit** et ainsi que **nomBanque** sont nécessaires pour effectuer un achat afin de pouvoir effectuer la vérification auprès de la banque. Il faut que le numéro de carte bancaire entré soit valide.

Les attributs issus de l'héritage sont déjà remplis puisque les contraintes précédentes ont été appliquées.

Placement des attributs

Ce sont des informations concernant chaque acheteur. Il faut pouvoir identifier et avoir les coordonnées de chaque acheteur, d'où l'héritage avec la super-classe **Compte**, pour les mêmes raisons que pour la sous-classe **Vendeur**. Un acheteur peut avoir plusieurs comptes/

Identifiant de la classe

numAcheteur est l'identifiant de la classe **CompteAcheteur** car lorsqu'un compte acheteur est créé, un numéro unique lui est attribué.

Choix d'une super-classe

Une classe association est adaptée aux différents statuts possibles du compte de l'acheteur. Un acheteur a un statut (très fidèle, fidèle, ...) lié au nombre d'achats qu'il a effectués.

Héritage

Cette classe association aura pour statut soit "assez fidèle", soit "fidèle", soit "très fidèle" et pas plusieurs en même temps, ainsi, on applique un héritage de *partition*.

Livre

Classe

Possède des informations sur les Livres soumis par les vendeurs.

Attributs

numLivre : INT NOT NULL
titre : VARCHAR NOT NULL
auteur : VARCHAR NOT NULL
catégorie : VARCHAR NOT NULL
edition : VARCHAR NOT NULL
format : VARCHAR NOT NULL
nbrPages : INT NOT NULL
dateParution : DATE NOT NULL < dateAnnonce DATE NOT NULL
etatGeneral : {neuf,TBE,BE,usé} NOT NULL
fourchettePrixMin : INT NOT NULL < fourchettePrixMax : INT NOT NULL
motClé1 : VARCHAR NOT NULL
motClé2 : VARCHAR
motClé3 : VARCHAR

Contraintes sur les attributs

numLivre NOT NULL car c'est l'identifiant de la classe qui permet d'identifier un livre.

Tous les attributs de cette classe ne peuvent pas être nuls puisque lors de la soumission d'un livre, le vendeur qui lui soumet est obligé de donner tous ces attributs. Cela ne s'applique pas pour les mots clés, il doit y en avoir au minimum un qui décrit le livre.

De plus, des attributs comme **titre**, **auteur**, **catégorie** et **motClé1** ne peuvent pas être nuls parce que lorsqu'un acheteur souhaite faire une recherche ou une réservation, il fournit ces attributs qui permettent de trouver le livre plus facilement.

En ce qu'il concerne **fourchettePrixMin** et **fourchettePrixMax**, une condition est appliquée, il faut que **fourchettePrixMin soit plus petit que fourchettePrixMax**. De plus, la **dateParution** ne doit pas être au futur, donc ne pas dépasser la date de publication du livre sur le site par le vendeur.

L'attribut **etatGeneral** doit être choisi entre **neuf**, **TBE**, **BE** et **usé**.

Placement des attributs

Tous ces attributs caractérisent un livre, c'est pourquoi ils sont dans la classe **Livre**.

Identifiant de la classe

numLivre n'est pas donné dans le sujet, pourtant c'est un attribut essentiel pour l'identification d'un livre publié sur le site.

Rechercher

Classe association / Super-classe

Permet à l'acheteur de rechercher un livre en vente par un vendeur.

Attributs

titre : VARCHAR

auteur : VARCHAR

catégorie : VARCHAR

Identifiants des classes Livre & CompteAcheteur :

numAcheteur : INT NOT NULL

numLivre : INT NOT NULL

Contraintes sur les attributs

Ces attributs n'ont pas de contraintes, il s'agit d'une recherche et tous les attributs ne sont pas obligatoires pour lancer la recherche d'un livre, il faut qu'il y en ait au moins un de rempli. *Les attributs issus de l'association sont déjà remplis puisque les contraintes précédentes ont été appliquées.* Si, à un moment donné, l'acheteur ne trouve pas de livre disponible correspondant au titre qu'il cherche, il peut faire une réservation.

Placement des attributs

Ces attributs sont dans la classe recherche puisque ce sont des éléments utiles à l'acheteur pour trouver un livre en particulier, que ce soit par la catégorie, son titre ou son auteur.

Identifiant de la classe

Il n'y a pas besoin d'identifier chaque recherche effectuée par l'utilisateur.

Choix d'une classe association

Une classe association est adaptée pour pouvoir entrer les champs de recherche de l'acheteur lors de sa navigation sur le site.

De **CompteAcheteur** à **Livre** il y a une multiplicité de minimum 1, car il y a un acheteur par livre, puis, de **Livre** à **CompteAcheteur** il y a une multiplicité de 1, parce qu'il peut y avoir qu'un acheteur par livre.

Choix d'une super-classe

Une super-classe est adaptée aux choix possibles de l'acheteur sur le site après avoir effectué une recherche.

Héritage

La classe association **Recherche** a pour héritage une *partition*, car l'Acheteur après sa recherche va soit consulter le livre pour mettre une évaluation et un commentaire, soit acheter directement un livre si il est disponible, soit il va réserver le livre en attendant qu'il soit disponible à l'achat.

Il ne peut pas acheter et réserver en même temps puisque si le livre n'est pas disponible il ne pourra pas l'acheter mais uniquement le réserver.

Consulter

Sous-classe de Rechercher

Résultats de la consultation d'un livre par un acheteur.

Attributs

dateConsultation : DATE NOT NULL

evaluation : INT : {1-10}

commentaire : VARCHAR

Attributs issus de l'héritage de Rechercher :

titre : VARCHAR

auteur : VARCHAR

catégorie : VARCHAR

numAcheteur : INT NOT NULL

numLivre : INT NOT NULL

Contraintes sur les attributs

L'attribut **dateConsultation** est obligatoire, car l'entreprise veut pouvoir récupérer cette donnée à des fins de statistiques. L'attribut **Evaluation** et commentaire ne sont pas obligatoires. L'évaluation laissée par l'acheteur sera comprise entre 1 et 10.

Les attributs issus de l'héritage sont déjà remplis, les contraintes précédentes ont été appliquées.

Placement des attributs

On retrouve les attributs de la super-classe **Rechercher**, qui permettent de caractériser la consultation, d'où la nécessité de pouvoir identifier le livre et l'acheteur. Les attributs **dateConsultation**, **evaluation** et **commentaire** sont propres à cette sous-classe **Consulter**.

Identifiant de la classe

La classe ne possède pas d'identifiant. C'est grâce à **numAcheteur**, **numLivre** et la **dateConsultation** que l'on peut identifier une consultation d'une autre.

Réserver

Sous-classe de Rechercher

Possède des informations sur une réservation d'un livre non disponible.

Attributs

dateReservation : DATE NOT NULL

Attributs issus de l'héritage de Rechercher :

titre : VARCHAR

auteur : VARCHAR

catégorie : VARCHAR

numAcheteur : INT NOT NULL

numLivre : INT NOT NULL

Contraintes sur les attributs

L'attribut **dateReservation** est obligatoire, car lorsqu'une personne réserve un livre et qu'il est disponible, la priorité sera donnée à la réservation la plus ancienne. Plusieurs acheteurs peuvent réserver le même livre. Le système enverra un mail à l'acheteur dès qu'il y aura un livre disponible.

Les attributs issus de l'héritage sont déjà remplis puisque les contraintes précédentes ont été appliquées.

Placement des attributs

La **dateReservation** est propre à cette sous-classe **Réserver**. On retrouve les attributs de la super-classe **Rechercher**, qui permettent de caractériser la réservation. De plus, le système enverra un mail à l'acheteur dès qu'il y aura un livre disponible, d'où la nécessité de pouvoir identifier le livre et l'acheteur.

Identifiant de la classe

La classe ne possède pas d'identifiant. C'est grâce à **numAcheteur**, **numLivre** et la **dateReservation** que l'on peut identifier une réservation d'une autre.

Achat

Sous-classe de Rechercher

Possède les informations sur l'achat d'un livre.

Attributs

Attributs issus de l'héritage de Rechercher:

titre : VARCHAR

auteur : VARCHAR

catégorie : VARCHAR

numAcheteur : INT NOT NULL

numLivre : INT NOT NULL

refAchat : VARCHAR NOT NULL

dateAchat : DATE NOT NULL

adresseLivraison : VARCHAR

Contraintes sur les attributs

Les attributs **refAchat** et **dateAchat** sont obligatoires. L'attribut **refAchat** aura une valeur unique qui permettra d'identifier les achats. L'attribut dateAchat est obligatoire parce qu'il est nécessaire pour déterminer le délai d'acceptation de l'achat et donc de l'évaluation du vendeur.

adresseLivraison n'est pas obligatoire, car l'acheteur doit indiquer l'adresse de livraison si elle est différente de l'adresse donnée lors de l'inscription.

Les attributs issus de l'héritage sont déjà remplis puisque les contraintes précédentes ont été appliquées.

Placement des attributs

On retrouve les attributs issus de l'héritage qui permettent d'avoir plus d'informations sur le livre acheté. refAchat et dateAchat sont des attributs propres à cette classe.

Identifiant de la classe

refAchat est l'identifiant de la classe, c'est un attribut dont la valeur est unique pour chaque achat et qui permettra d'identifier facilement un achat.

Accepter

Classe association

Le vendeur doit accepter un achat effectué par un compte acheteur.

Attributs

Attributs issus de l'association entre Vendeur et Achat :

numAcheteur INT NOT NULL

refAchat INT NOT NULL

dateAcceptation : DATE

dateLivraisonEstimée : DATE > dateAcceptation

FourchettePrixMin < prix INT < FourchettePrixMax

Contraintes sur les attributs

Les attributs **dateAcceptation**, **dateLivraisonEstimée** et **prix** doivent obligatoirement être complétés si le vendeur à accepter la vente ! Autrement, ils ne sont pas obligatoires s'il y a refus du vendeur.

Dans les deux cas, par cette classe association entre **Achat** et **Vendeur**, la règle de transitivité s'applique et nous accédons à **numLivre** pour savoir quel livre est sur le point d'être acheté, et surtout à quelle fourchette de prix le vendeur peut fixer son prix, d'où la condition **FourchettePrixMin < prix < FourchettePrixMax**.

La date de livraison estimée sera forcément supérieure à la date d'acceptation.

Les attributs issus de l'association sont déjà remplis puisque les contraintes précédentes ont été appliquées.

Placement des attributs

Ces attributs concernent l'acceptation du vendeur lorsqu'un achat est proposé à un certain prix. Nous avons besoin de récupérer la donnée sur la date d'acceptation pour ensuite déterminer la note du vendeur dans la classe **Evaluation**. Ensuite, le prix est fixé par le vendeur lors de l'acceptation, et un mail est envoyé à l'acheteur pour le prévenir de la date de livraison estimée.

Nous savons à quel achat et vendeur s'applique l'acceptation grâce à l'association, donc la classe possède les identifiants de ces classes.

Identifiant de la classe

Cette classe n'a pas d'identifiant propre, elle prend les identifiants des classes auxquelles elle est associée.

Choix d'une classe association

Il faut pouvoir modéliser l'acceptation du vendeur sur l'achat. Donc en associant l'achat et le vendeur, on récupère les dates et le prix fixé par le vendeur.

D'**Achat** à **Vendeur** et de **Vendeur** à **Achat**, il y a multiplicité à 1. Un vendeur ne peut accepter qu'un achat à la fois, et chaque achats a un vendeur à la fois.

Evaluation

Super-classe

Possède des informations sur l'évaluation d'un vendeur

Attributs

$\text{delaiAcceptation} : \text{INT NOT NULL} = \text{dateAchat} : \text{INT NOT NULL} - \text{DateAcceptation} : \text{INT NOT NULL}$

Contraintes sur les attributs

L'attribut **delaiAcceptation** ne peut pas être vide, c'est l'élément qui va stocker le résultat du calcul de la note du vendeur.

D'après la règle de transitivité, l'association relie les classes **Achat**, **Accepter** et **Vendeur** à **Evaluation**. Le vendeur auquel l'évaluation est appliquée est identifié par **numVendeur**. Cette note sera calculée à partir de l'attribut **dateAchat**, provenant de la classe **Achat**, et de la **dateAcceptation**, provenant de la classe **Accepter**.

Placement des attributs

La classe évaluation évalue un vendeur, et l'évaluation est calculée à partir du délai d'acceptation d'une vente, c'est pourquoi on place **delaiAcceptation** dans la classe, et les attributs provenant de la classe **Accepter** et **Achat**.

Identifiant de la classe

La classe association **Evaluation** n'a pas d'identifiant. Elle se base sur les identifiants des classes avec lesquelles la transitivité s'applique.

Choix d'une classe association

Une super-classe se porte mieux au contexte de la notation du vendeur, car cette notation peut avoir 5 états.

Héritage

La classe "évaluation" a pour héritage une partition, car chaque vendeur a une évaluation 1 à 5 étoiles, mais pas deux types de notes en même temps.

Associations

publie / est publié

Nom : publie de vendeur à livre / est publié de livre à vendeur

Description : Cette association représente la publication de la vente d'un livre par un compte vendeur, notamment par le remplissage du formulaire sur le site web.

Le vendeur peut publier un à autant de livre qu'il le souhaite, le vendeur est lié au minimum à un livre.

Chaque livre est publié par un vendeur à la fois, donc un livre est lié au minimum à un et maximum à un vendeur .

évalue / est évalué

Nom : "évalue" d'évaluation à vendeur / "est évalué" de vendeur à évaluation

Description : Cette association représente l'évaluation d'un compte vendeur, en fonction de son délai d'acceptation de l'achat, par un compte acheteur, d'un livre qu'il a publié.

Le vendeur peut avoir une note à la fois, donc le vendeur est lié au minimum à une et maximum à une évaluation.

Il y a une évaluation par vendeur, donc un livre est lié au minimum à un et maximum à un vendeur.

permet

Nom : "permet" de réservation à achat

Description : Cette association représente la permission d'achat après avoir fait une réservation

Une réservation peut mener à l'achat d'aucun livre et d'un livre au maximum, car on suppose que l'acheteur n'est pas obligé d'en acheter après avoir réservé, ou qu'un autre acheteur avec une date de réservation plus ancienne va l'acheter une fois disponible.

