



Projet BD - Réalisation d'une application de commerce

Documentation Du Projet

Fait par les étudiants

Zineb Et-tarraf
Nada Boukhriss
Safia Echarif
Youssef Benhachem
Reda Achouhad

Novembre 2021

Abstract

Dans ce projet, On travaille sur la réalisation de l'application Gange afin de mettre à la vente les produits de Cette société de commerce. On documente donc dans ce document tout le travail fait en détails, depuis l'analyse du problème, à la conception de la base de données et sa traduction en relationnel, puis l'implémentation de l'application en JAVA en utilisant la bibliothèque graphique Swing, en donnant une attention particulière aux fonctionnalités.

1 La Conception de Bases de Données

1.1 L'Analyse Statique du Problème:

Après la lecture et l'analyse de la description de l'application et des fonctionnalités demandées dans l'application, on a finalement conclu un ensemble de types-entités qui seraient constructives du système d'informations. On en cite :

- **Produit** : Qui est le produit mis à la vente. Il est caractérisé par : Son identifiant, Son intitulé, Son Prix Courant, Une petite description, LA (seule) catégorie à laquelle il appartient, un ensemble de caractéristiques et valeurs, et enfin un lien URL vers son image.
- **Client** : Qui est le client qui utilise notre application pour enchérir. C'est une personne réelle et concrète qui enchérit. Il est caractérisé par : Son mail, Son Mot de Passe, Son Nom, Son Prénom et son Adresse Postale.
- **Utilisateur** : C'est le statut de la personne qui sert de client au sein de l'application. Ce statut permet de garder une certaine cohérence de la base quand un client décide d'utiliser son droit à l'oubli.
- **Offre** : C'est l'offre que le client fait sur un produit dans l'espoir de l'acheter. Elle est caractérisée par la date et heure de son émission et le prix proposé.
- **Achat** : La 5e et dernière offre sur un certain produit, et qui permet au client de l'Obtenir.
- **Catégorie** : Qui englobe l'ensemble des "types" ou "genres" de nos produits... et qui est assez essentielle dans le parcours des produits offerts.

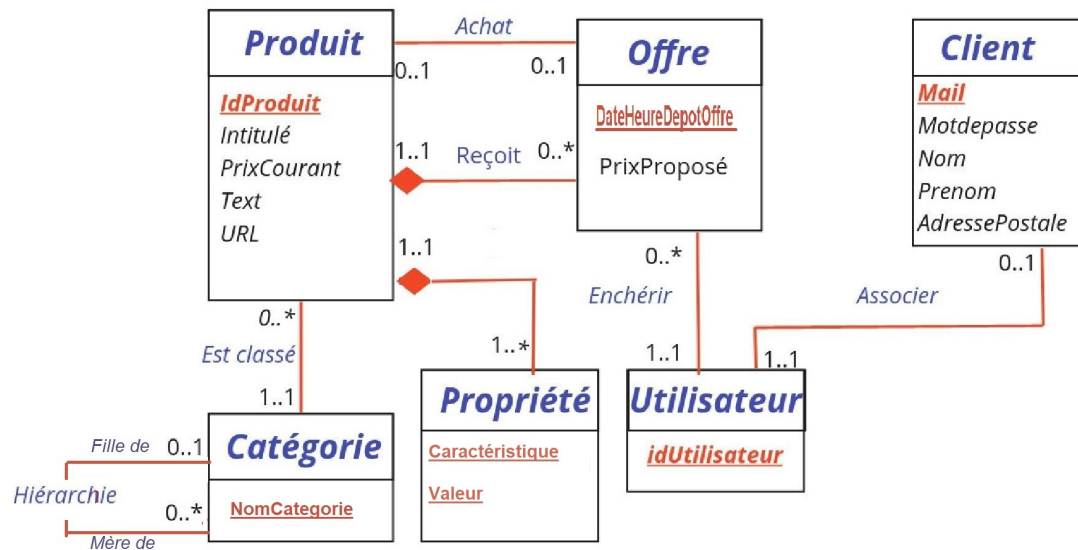
Ensuite en analysant davantage les propriétés de ces entités, on a conclut l'existence d'autres entités qui ne seraient là que pour les qualifier, comme:

- **Propriété** : qui note le couple Caractéristique/Valeur qui caractérise un produit.
- **Hierarchie** : qui lie chaque catégorie fille à sa catégorie mère.

Dépendance Fonctionnelle	Contraintes de Valeur	Contraintes de multiplicité	Contraintes Contextuelles
- Idproduit → Intitulé, PrixCourant, description, url - Idproduit → NomCategorie - mail → MotDePasse, NomClient, PrenomClient, AdressePostale - mail → idUtilisateur - idproduit, dateheureDepotOffre → prix proposé, idUtilisateur (Sur le même produit deux offres ne peuvent être faites au même instant.)	- PrixCourant > 0 - PrixPropose > PrixCourant (Principe d'enchères)	- Idproduit -->> caractéristique, valeur (' Un produit admet au moins un couple caractéristique/valeur') - idUtilisateur -> mail (Un utilisateur peut être un client ou non) - idUtilisateur -->> dateheureDepotOffre, IdProduit (Un utilisateur peut faire des offres de prix à préciser sur un produit) - Catégorie -->> CatégorieFille (Chaque Catégorie peut avoir des sous-catégories)	- Le prix proposé doit être toujours supérieur au prix courant - La cinquième offre emporte l'enchère - Les recommandations concerneront les catégories pour lesquelles l'utilisateur a fait le plus d'offres sans remporter aucune (Recommandations Personnelles), ensuite les catégories qui ont le plus d'offres en moyenne (Tendances). - Le Client a le droit à l'oubli, il peut supprimer son compte.

1.2 Conception Entités/Associations

Après Plusieurs essais, discussions et analyse, nous avons conclu Le Schéma Entité/Associations suivant:



Son Explication :

1. Association entre utilisateur et client:

- Cardinalité 0..1 : d'après la contrainte de multiplicité, on a la dépendance fonctionnelle $idUtilisateur \rightarrow mail$ est donc 'Un utilisateur peut être un client ou non'. En plus on a le principe de 'de droit de l'oubli', si un client veut supprimer ses données personnelles, l'application va créer automatiquement un compte utilisateur avec un nouvel idutilisateur associé aux offres et achats de ce clients. Ce nouveau compte n'est pas forcément associé à un client.
- Cardinalité 1..1 : chaque client est forcément un utilisateur. Chaque client doit avoir un idutilisateur.

2. Association entre utilisateur et Offre

- Cardinalité 0..* : Un utilisateur peut faire une ou plusieurs offres de prix sur différents produits, comme il peut ne pas en faire.
- Cardinalité 1..1 : Une offre est nécessairement faite par un seul utilisateur.

3. Associations entre offre et produit

(a) Association reçoit:

- Cardinalité 0..* : Plusieurs offres peuvent étre faites sur le même produit (Le nombre des offres ne dépasse pas 5 bien sûr, mais ceci est géré par l'application directement)
- Cardinalité 1..1 : Une offre ne peut concerner qu'un et un seul produit.
- Une Offre est nécessairement liée à un produit. Son existence dépend bien donc du produit auquel elle est liée... Elle en est donc une entité faible

(b) Association Achat: L'achat est matérialisé en enregistrant la correspondance entre le produit et l'offre gagnante.

- Cardinalité 0..1 : Une Offre peut être gagnante ou pas
- Cardinalité 0..1 : Un produit peut être acheté ou pas

4. Association entre produit et catégorie

- Cardinalité 0..* : Une Catégorie peut contenir un ou plusieurs produits ou pas
- Cardinalité 1..1 : un produit est classé dans une seule catégorie

5. Association entre produit et propriété

- Cardinalité 1..* : Un produit possède une ou plusieurs propriétés
- Cardinalité 1..1 : Une propriété correspond bien à un produit.

L'entité propriété est bel est bien une entité faible de produit, puisque une propriété n'aura aucun sens si elle n'est pas associée à un produit donné.

6. Association entre catégorie et elle même

- Cardinalité 0..1 : toutes les catégories ont une catégorie mère, sauf la racine.
- Cardinalité 0..* : Une catégorie mère peut avoir une ou plusieurs catégories filles ou peut rien avoir.

2 La Traduction au relationnel

2.1 1ère étape : Les Entités Simples

On a commencé par repérer les entités simples, en créant pour chacune une relation explicitant ses attributs, et dont la clé sera les attributs clés de l'entité.

- Produit (***IdProduit***, Intitulé, PrixCourant, Description, URL)
- Client (***Mail***, MotDePasse, Nom, Prénom, AdressePostale)
- Utilisateur (***IdUtilisateur***)
- Catégorie (***NomCatégorie***)

2.2 2e étape : Les Entités Faibles

Ensuite, on a examiné les entités faibles :

- On a l'entité Offre est une entité faible par rapport à l'entité Produit, donc par les règles de traduction la clé primaire de l'entité offre sera DateHeureDepotOffre , IdProduit, la relation créée sera :
Offre (***DateHeureDepotOffre***, ***IdProduit***, PrixProposé)
- Propriété est une entité faible de Produit, On ajoute donc les attributs clefs de la relation Produit aux attribut de la relation Propriété, comme suit :
Propriété(***Caractéristique***, ***Valeur***, ***IdProduit(étrangère)***)

2.3 3e étape : Les Associations:

1. Associations de Cardinalité (1..1)

- Pour l'association produit / catégorie, on la cardinalité (1..1) se trouve à côté de Catégorie , donc on ajoute l'attribut NomCategorie (qui est la clé primaire de l'entité Catégorie) comme clé étrangère dans la relation Produit. On obtient donc la relation Produit comme suit:
Produit (***IdProduit***, Intitulé, PrixCourant, description, URL, NomCatégorie (étrangère))
- L'association Client / Utilisateur est une association binaire dont la cardinalité (1..1) est du côté de l'entité Utilisateur, par conséquent on intègre dans la relation Client la clef de la relation Utilisateur, comme suit :
Client (***Mail***, MotDePasse, Nom, Prénom, AdressePostale, IdUtilisateur (étrangère))

- L'association Offre / Utilisateur est une association binaire dont la cardinalité (1..1) est du côté de l'entité Utilisateur, ainsi on ajoute dans la relation Offre la clef de la relation Utilisateur, comme suit :
Offre (*DateHeureDepotOffre*, *IdProduit (étrangère)*, PrixProposé, IdUtilisateur (étrangère))

2. Associations de cardinalité (0..1)

- L'association Produit / Offre est une association binaire dont les deux cardinalités sont (0..1), on crée une nouvelle relation Achat en regroupant les clefs de Produit et Offre comme des clefs étrangères et primaires de Achat, du fait que la cardinalité (0..1) est du côté du Produit et Offre.
Achat(*IdProduit*, *DateHeureDepotOffre*)
- L'association Catégorie / Catégorie : L'association Catégorie / Catégorie est une association dont une deux cardinalités sont (0..1), donc on crée une nouvelle relation Hiérarchie comme suit :
Hiérarchie(*NomMère*, *NomFille*)

3. Associations de cardinalité (0..*)

- Toutes les associations de cardinalité (0..*) ont été déjà traité.

2.4 Schéma Relationnel final

Suite à la traduction, on obtient le schéma relationnel suivant :

- Produit (*IdProduit*, Intitulé, PrixCourant, Description, URL, NomCatégorie(étrangère)).
- Propriété(*Caractéristique*, *Valeur*, *IdProduit(étrangère)*).
- Offre (*DateHeureDepotOffre*, *IdProduit (étrangère)*, PrixProposé, IdUtilisateur (étrangère)).
- Achat (*IdProduit*, *DateHeureDepotOffre*).
- Utilisateur (*IdUtilisateur*) .
- Client (*Mail*, MotDePasse, Nom, Prénom, AdressePostale, IdUtilisateur(étrangère)).
- Catégorie (*NomCatégorie*).
- Hiérarchie(*NomMère*, *NomFille*).

2.5 Vérification des Formes Normales

On vérifie que toutes nos relations sont 3FNBCK pour qu'on aie pas de redondance dans notre schéma.

Les relations : Propriété, Achat, Utilisateur, Catégorie et Hiérarchie n'ont que des attributs clefs donc ils sont tous des 3FNBCK, donc on ne fera la vérification que sur les tables : Produit, Client et Offre .

1. Vérification 1FN

- Pour la relation Produit : tous ses attributs sont atomique (ne sont pas des listes d'éléments).
- Pour la relation Client : tous ses attributs sont atomique (ne sont pas des listes d'éléments).
- Pour la relation Offre : tous ses attributs sont atomique (ne sont pas des listes d'éléments).

2. Vérification 2FN

Tous ces relations sont 1FN et de plus :

- Pour la relation Produit : tous les attributs non-clefs de la table Produit sont pleinement dépendants de la clef IdProduit. En effet :

- IdProduit \rightarrow Intitulé.
- IdProduit \rightarrow PrixCourant.
- IdProduit \rightarrow Description.
- IdProduit \rightarrow URL.
- IdProduit \rightarrow NomCatégorie.
- Pour la relation Client : tout les attributs non-clefs de la table Client sont pleinement dépendants de la clef Mail. En effet :
 - Mail \rightarrow MotDePasse.
 - Mail \rightarrow Nom.
 - Mail \rightarrow Prénom.
 - Mail \rightarrow AdressePostale.
 - Mail \rightarrow IdUtilisateur.
- Pour la relation Offre : tout les attributs non-clefs de la table Offre sont pleinement dépendants des attributs clefs (DateHeureDepotOffre, IdProduit). En effet :
 - DateHeureDepotOffre, IdProduit \rightarrow PrixProposé.
 - DateHeureDepotOffre, IdProduit \rightarrow IdUtilisateur.

3. Vérification 3FN

Tous ces relations sont 2FN et de plus, pour chaque relation : aucun attribut non clef ne dépend d'un autre attribut non-clef.

4. Vérification 3FNBCK

Comme tous ces relations sont 3FN on a de plus, pour chaque relation : toutes ses dépendances fonctionnelles non triviale contiennent une clef de cette relation .

3 Les Fonctionnalités

3.1 L'Analyse des Fonctionnalités

1. le Parcours du Catalogue

- La connexion du client : le client a le choix de se connecter à l'application si il a déjà un compte, ou à s'enregistrer. Si par hasard, le client essaie de se connecter avec un mail déjà existant sur notre base de donnée on affiche un message d'erreur.
- Parcours des catégories et sous catégories ou catégories recommandées :
 - Parcours des catégories et sous catégories :
le client choisit de parcourir la liste des produits d'une catégorie quelconque : si il choisit de parcourir une catégorie fille on affiche juste les produits de celle-ci, sinon, si il choisit de parcourir le catalogue d'une catégorie mère, on affichera tous les produits de toutes les catégories filles de cette catégorie mère. et par ordre alphabétique
 - Parcours des catégories recommandées :
Le client à le choix de parcourir les recommandations personnelles ou les tendances.
 - * Parcours des recommandations personnelles :
Au début on avait eu l'idée d'étendre les recommandations personnelles à toutes les catégories soeurs immédiates des catégories sur lesquelles l'utilisateur enchérit sans les acheter... Ceci était dans le but d'avoir un panel de recommandations assez large. Et pour le faire en SQL on avait envisagé l'utilisation de vues (La vue MostBidOn qui est l'ensemble des catégories sur lesquelles le client a enchérit sans achat, puis celle de leurs mères et enfin recommander toutes les catégories filles de ces mères qui seront donc les soeurs immédiates des catégories de la vue MostBidOn...). Par contre on a vite abandonné l'idée quand on a découvert que parmi ces soeurs immédiates il peut y avoir des catégories parmi lesquelles il a déjà effectué des achats. Et donc On ne respecterait plus le cahier de charges. Ainsi,

on se contente d'afficher les produits des catégories (juste les catégories filles sans remonter aux catégories mères) sur lesquelles l'utilisateur a fait le plus d'offres sur des sans réussir à les acheter. Si l'utilisateur n'a pas fait encore des offres, on affiche une page vide sans produits.

* Parcours des tendances :

On affichera les produits des catégories (juste les catégories filles) sur lesquelles il y a eu le plus d'offres en moyenne par produit.

L'affichage des produits se fera toujours par ordre croissant d'offres qui ont été faites sur ces produits.

- Consultation de la fiche complète d'un produit sélectionnés :

2. les Enchères

Une fois qu'un Client décide d'enchérir sur un produit et nous propose un prix d'achat, on compare d'abord son prix proposé au prix courant du produit, puis on s'assure que le produit n'a pas été acheté. Ainsi, si l'offre est acceptée, on vérifie si c'est une offre gagnante ou pas en évaluant le nombre de gens qui ont déjà enchéri sur le produit. Si c'est une offre ordinaire, on se contente d'insérer les informations liées à cette nouvelle offre dans notre table OFFRE et de mettre à jour le prix courant du produit concerné dans la table OFFRE. Si de plus l'offre émise est gagnante, on insère aussi les informations liées à cette offre dans la table ACHAT pour le matérialiser.

Un des problèmes qu'on a rencontré lors de l'implantation de cette dernière fonctionnalité (achat), était le fait qu'en utilisant le mot clé CURRENT TIMESTAMP pour insérer l'offre gagnante dans Achat, la date aurait forcément changé aux milisecondes depuis la date où ça a été inséré dans offre, et du coup il ne retrouve plus cette date dans la table OFFRE. Et après quelques tentatives d'essais de détection du problème, on a finalement décidé de récupérer la date directement de la table Offre (Requêtes SQL dans La partie implantation)

On a aussi choisi de gérer les transactions dans cette partie en assurant la sérialisabilité, et en désactivant l'autocommit.

3. Droit à l'oubli

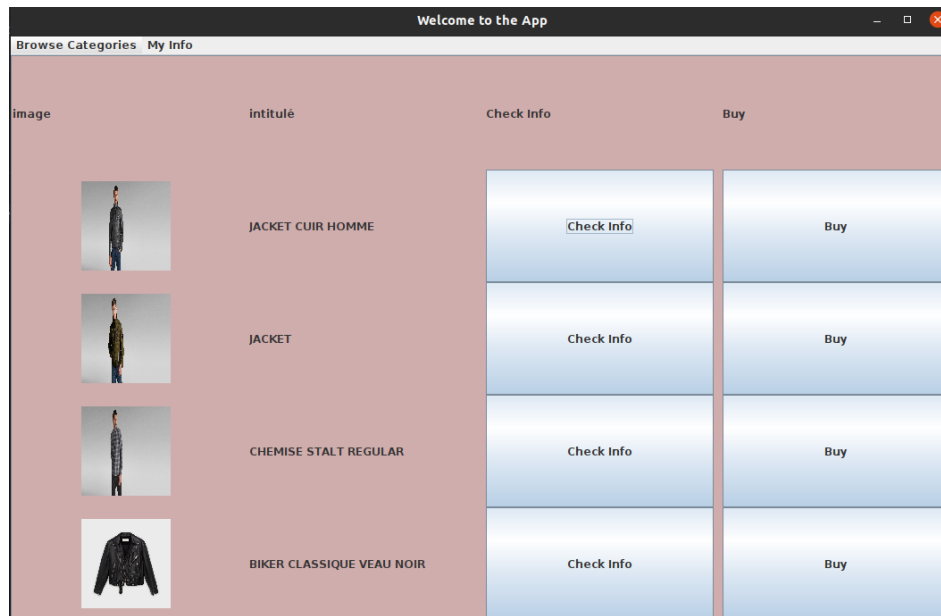
Une personne peut toujours renoncer à la création de son compte. Dès lors, on donne aux clients la main pour effacer leurs données personnelles. Pour ce faire, le client peut cliquer sur le bouton "delete account" qui supprimera définitivement ses données de la table client, et qui changera son Identifiant dans les autres tables (Achat et offre pour qu'on perd pas les informations de notre application).

4 Mode d'Emploi de l'Application

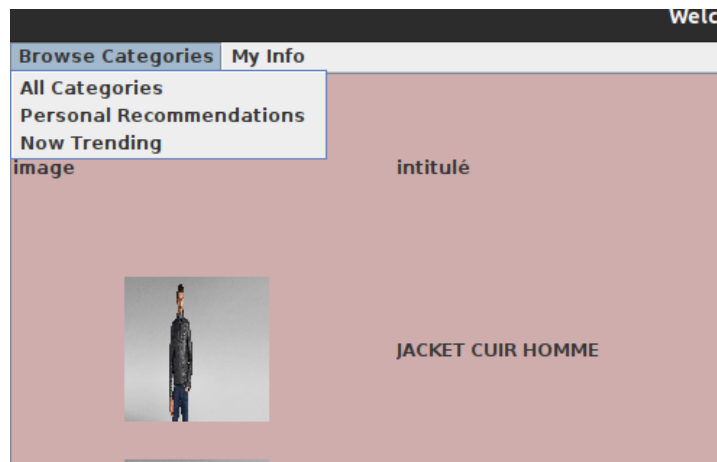
Il suffit de compiler les fichiers .java, exécuter le fichier Gange.java qui affichera l'application (une interface graphique). Vous pouvez donc soit vous connectez soit vous enregistrez sur l'application.



Lors de la connexion, l'application affiche en premier des produits aléatoires, affiche également des boutons qui permettent de choisir entre les fonctionnalités :



- browse categories :
 - all categories.
 - Personal recommendations.
 - New trending.



- My info :

Name	Comoth
Last Name	Antoin
Adresse Mail	antoin@gmail.com
Mot de Passe	123456789
Adresse Postale	38 Paris
Delete Account	Delete Account

5 Bilan du Projet

En conclusion, ce projet a été très instructif pour nous. Ce fut une très belle opportunité de comprendre de manière très pratique et réelle la conception et le fonctionnement d'une base de données afin de répondre à une problématique, bien au-delà de ce que l'on pouvait apprendre à travers des enseignements théoriques. C'était aussi une belle opportunité d'appliquer toutes les notions apprises en classe, et d'acquérir de nouvelles compétences comme gérer les connexions à la base de données et travailler sur l'interface utilisateur (même si ce n'était point le but du projet).