Ensimag 2A IF

PROJET DE BASE DE DONNÉES

Documentation

3 décembre 2024

ADNET Willem (adnetw)
BADIANE Seydina (badianes)
BARKATOU Walid (barkatow)
MOUGANG TCHAMINI Megane (mougangp)
ZABIÉGO Hugo (zabiegoh)
ZINE Ouadii (zineo)

Sommaire

1 INTRODUCTION **3**

2 Analyse du problème **3**
2.1 Entites et Propriétés élémentaires 3
2.2 Contraintes 4
2.3 Modèle Entités/Associations 6
2.4 Explication des Choix 6

3 Traduction en Modèle Relationnel **7**
3.1 Traduction des entités simples 7
3.2 Type d'entité faible 7
3.3 Types d'associations 7

4 Analyse des Fonctionnalités **8**
4.1 Mise en place d'une salle de vente 8
4.2 Placement d'une enchère 8
4.3 Processus de fin d'enchère 9
4.4 Mise à jour des stocks après une vente 9

5 Mode d'emploi du demonstrateur **10**

6 Bilan du projet **11**
6.1 Retour d'Expérience 11
6.2 Perspectives d'Amélioration 11

1 INTRODUCTION

L'objectif de ce projet était de développer une application pour la société Baie-électronique, qui souhaite informatiser son service de ventes aux enchères. Le projet a été structuré en plusieurs étapes clés. La première étape a consisté en la modélisation du problème, où nous avons analysé les besoins et les contraintes afin de créer un schéma Entités/Associations. Ce schéma représente les données nécessaires à l'application et leurs relations sémantiques. Ensuite, nous avons procédé à l'implantation de la base de données. Cela a impliqué la traduction du schéma conceptuel en un schéma relationnel, qui a ensuite été implanté sur le SGBD Oracle. Cette étape a permis de définir les structures nécessaires pour stocker les informations relatives aux produits, aux utilisateurs et aux enchères. L'étape suivante a été l'analyse des fonctionnalités. Nous avons élaboré les requêtes SQL nécessaires à la gestion des enchères, des salles de vente et des utilisateurs, tout en regroupant ces requêtes en transactions pour garantir la cohérence de la base de données, même en cas d'accès concurrents. Enfin, nous avons développé un démonstrateur en Java utilisant JDBC pour interagir avec la base de données. Ce démonstrateur permet aux utilisateurs de naviguer dans l'application, de placer des enchères et de visualiser les produits disponibles. Ce rapport détaille chacune de ces étapes, en expliquant les choix faits, les défis rencontrés, ainsi que les solutions apportées.

2 Analyse du problème

2.1 Entites et Propriétés élémentaires

Dans le cadre de notre projet, nous avons identifié plusieurs entités essentielles, chacune avec ses attributs spécifiques. Voici la liste des entités et leurs attributs :

- **Utilisateur**
  - *Email* (PK) : Identifiant unique de l'utilisateur.
  - *Nom* : Nom de l'utilisateur.
  - *Prenom* : Prénom de l'utilisateur.
  - *AdressePostale* : Adresse postale de l'utilisateur.
- **Produit**
  - *IdProduit* (PK) : Identifiant unique du produit.
  - *NomProduit* : Nom du produit.
  - *PrixRevient* : Prix de revient à partir duquel le vendeur réalise des bénéfices.
  - *Stock* : Quantité de produit disponible à la vente.
  - *DispoProduit* : Indicateur de disponibilité du produit.
- **Caractéristiques**
  - *NomCar (PK)* : Nom de la caractéristique du produit.
  - *ValeurCar* : Valeur associée à la caractéristique.
- **Vente**
  - *IdVente* (PK) : Identifiant unique de la vente.

  - *PrixDepart* : Prix de départ de la vente.
  - *PrixActuel* : Prix actuel du produit lors de la vente.
  - *Durée* : Durée de la vente, indiquant si elle est limitée ou libre.
  - *Quantité* : Quantité de produit concernée par la vente.
  - *DateVente* : Date à laquelle la vente a lieu.
- **SalleDeVente**
  - *IdSalle* (PK) : Identifiant unique de la salle de vente.
  - *NomSalle* : Nom de la salle de vente.
  - *EstOccupé* : Indicateur de disponibilité de la salle (occupée ou libre).
  - *EstMontante* : Indicateur du type de vente (montante ou descendante).
  - *EstRevocable* : Indicateur de la possibilité d'annuler la vente.
  - *LimiteOffre* : Limite le nombre d'offres.
  - *TypeDurée* : Type de durée de la vente (limitée ou libre).
- **Catégorie**
  - *NomCat* : Nom de la catégorie à laquelle appartient le produit.
  - *DescCat* : Description de la catégorie.
- **Offre**
  - *PrixOffre* : Montant de l'offre faite par l'utilisateur.
  - *DateOffre* : Date à laquelle l'offre a été faite.
  - *HeureOffre* : Heure à laquelle l'offre a été faite.
  - *Quantite* : Quantité de produit concernée par l'offre.

2.2 Contraintes

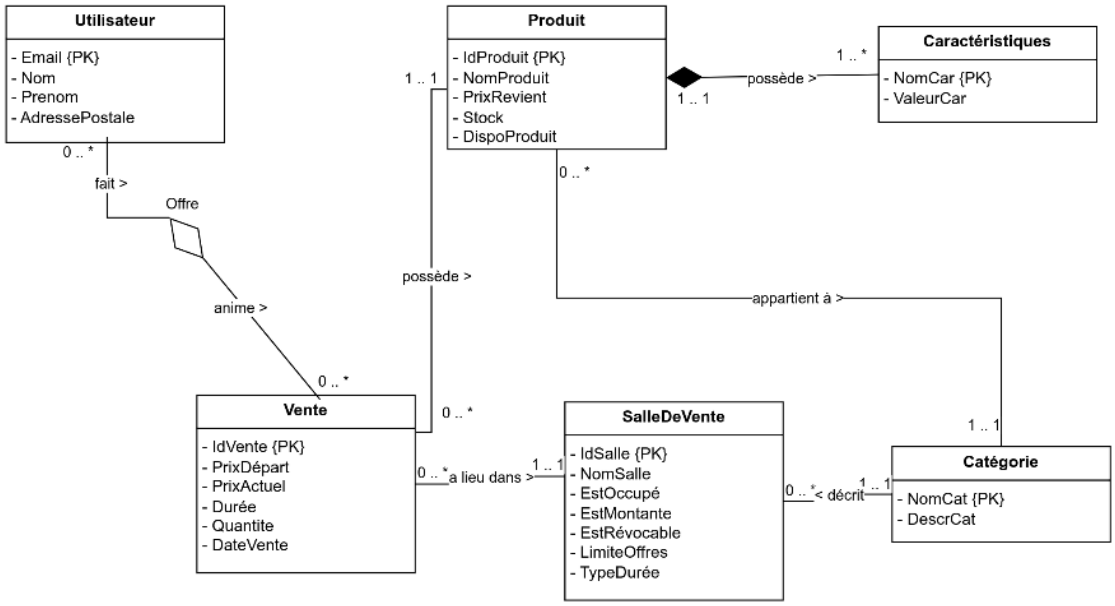
Après analyse du texte, nous avons relevé les contraintes suivantes :

Dépendances Fonctionnelles	**Contraintes de Valeur**	**Contraintes de Multiplicité**	**Contraintes Contextuelles**
`email` $\rightarrow$ `nom, prénom, adressepostale`	N/A	Un utilisateur peut faire 0 ou plusieurs offres : `email` $\nrightarrow\!\!\!\!\rightarrow$ `offre`	N/A
`idproduit` $\rightarrow$ `nomproduit, prixdepart prixrevient, stock`	`prixrevient` doit être positif. `stock` doit être ≥ 0.	N/A	La somme des quantités des offres gagnantes ne peut pas dépasser le stock du produit.
`nomcat` $\rightarrow$ `descrcat`	N/A	Une catégorie peut regrouper plusieurs produits : `nomcat` $\twoheadrightarrow$ `idproduit`	N/A
`idsalle` $\rightarrow$ `typedevente, catégorie, limiteoffres, estmontante, estoccupee, estrevocable, typeduree`	`limiteoffres` > 0 ou $= -1$; `typeduree` doit être `limitée` ou `illimitée`.	Une salle de vente peut contenir 0 ou plusieurs ventes : `idsalle` $\nrightarrow\!\!\!\!\rightarrow$ `idvente`	Une salle ne peut regrouper que des produits d'une catégorie.
`idvente` $\rightarrow$ `typevente, prixdépart, révocable, limitateoffres, durée, idproduit, idsalle`	`prixdépart` > 0.	Une vente peut recevoir 0 ou plusieurs offres : `idvente` $\nrightarrow\!\!\!\!\rightarrow$ `offre`	Les ventes peuvent être montantes ou descendantes, révocables ou non révocables. Pour une vente montante, les offres doivent être croissantes. Pour une vente descendante, le prix diminue régulièrement. Une vente ne peut avoir lieu que si le stock du produit est $>$ 0.
`idproduit, nomcar` $\rightarrow$ `valeurcar`	N/A	Plusieurs caractéristiques peuvent être associées à un produit : `idproduit` $\twoheadrightarrow$ `nomcar, valeurcar`	Les caractéristiques définissent les propriétés spécifiques d'un produit.

TABLE 1 – Tableau des éléments d'analyse

2.3 Modèle Entités/Associations

On obtient ensuite le diagramme Entité/Associations suivant.



Précision à apporter : Offre est la table issue de l'association 0..* entre Utilisateur et Vente. Aussi, Les attributs "EstMontante", "EstRevocable", "LimitesOffres" et "TypeDuree" ont été mis dans SalleDeVente pour garantir d'avoir un seul type de vente dans une salle. Ceci permet d'avoir des salles de vente specifiques.(ex : la salle des ventes montantes revocables sans limite d'offres et a duree illimitee, la salle des ventes descendantes revocables de limite=2 et avec 30 minutes de durée...).

2.4 Explication des Choix

Dans cette section, nous détaillons certains choix de conception effectués lors de la modélisation de la base de données pour le projet de gestion des enchères.

— Le choix d'inclure l'attribut **DispoProduit** dans l'entité Produit permet de gérer efficacement la disponibilité des produits tout au long des enchères.

— Nous avons décidé de concevoir des **Salles de Vente** spécifiques, qui ne contiendront qu'un seul type de vente à la fois. Cela permet de simplifier la gestion des ventes et de garantir les règles de chaque type de vente. Les attributs tels que *EstOccupé* et *EstMontante* permettent de gérer les états de la salle de vente. Ainsi, un utilisateur peut faire des offres d'un meme type et une vente d'un type deja defini dans une salle de vente donnée.

— Nous avons choisi de mettre l'entité **Caracteristiques** comme un composant de **Produit** car Cela permet d'ajouter facilement de nouvelles caractéristiques sans modifier la structure de Produit.

3 Traduction en Modèle Relationnel

Nous avons appliqué l'algorithme vu en cours pour obtenir le schéma relationnel.

3.1 Traduction des entités simples

Nous avons identifié les entités simples suivantes :

- **UTILISATEURS**(Email, Nom, Prénom, AdressePostale)
- **PRODUITS**(IdProduit , NomProduit, PrixRevient, Stock, DispoProduit, **NomCat**)
- **VENTES**(IdVente, PrixDepart, PrixActuel, Quantité, DateVente, **IdSalle**, **IdProduit**)
- **SALLEDEVENTE**(IdSalle (pk), NomSalle, EstOccupé, EstMontante, TypeDurée, **NomCat**)
- **CATEGORIE**(NomCat, DescCat)

3.2 Type d'entité faible

L'entite "Caracteristiques" a une relation de composition avec "Produit". Caracteriques est un composant de Produit.

- **CARACTERISTIQUES**(NomCar, **IdProduit**, ValeurCar)

3.3 Types d'associations

Les attributs en gras dans les relations de la partie 3.1 correspondent aux traductions des associations avec cardinalité 1..1.

En ce qui concerne les relations de cardinalités 0..*, nous avons la relation entre "Utilisateur" et "Vente" qui conduit a la creation de la nouvelle table **Offre**.

- **OFFRES**(PrixOffre, DateOffre, HeureOffre, Quantité, **Email**, **IdVente**)

Toutes les relations sont 3FNBCK . Tout attribut non clef est pleinementet directement-dépendant de toutes les clés(ici on a une unique clé à chaque fois). Ce qui assure la 3FN. En plus de cela pour toute DF X–>Ynon trivial, X contient une clef de la relation. D'où la forme 3FNBCK.

4 Analyse des Fonctionnalités

Elle se concentre sur les transactions essentielles. Chaque fonctionnalité est décrite avec son implantation sous forme de requêtes SQL, accompagnées de commentaires explicatifs.

4.1 Mise en place d'une salle de vente

La mise en place d'une salle de vente est cruciale pour organiser les enchères. Voici comment cette fonctionnalité est implémentée :

```
-- ============================
-- Création d'une nouvelle salle de vente
-- ============================
BEGIN;

-- Insertion des détails de la salle de vente
INSERT INTO SALLEDEVENTE (IDSALLE, ESTMONTANTE, ESTOCCUPEE, ESTREVOCABLE,
LIMITEOFFRES, TYPEDUREE, CATEGORIE)
VALUES ('7', '1', '0', '1', '12', 'illimitee', 'Livres');

COMMIT;
```

4.2 Placement d'une enchère

Le placement d'une enchère permet aux utilisateurs de soumettre leurs offres sur un produit. Voici l'implémentation :

```
-- ============================
-- Placement d'une enchère
-- ============================
BEGIN;

-- Insertion d'une nouvelle offre
INSERT INTO OFFRE (PrixOffre, HeureOffre, Quantite, Email, IdVente)
VALUES ('95', CURRENT_TIMESTAMP, '1', 'jennifer.wilson@bdd.com', '6');

-- Mise à jour du prix actuel de la vente
UPDATE VENTE
SET PrixActuel = '95'
WHERE IdVente = '6';

COMMIT;
```

4.3 Processus de fin d'enchère

Le processus de fin d'enchère est essentiel pour déterminer le gagnant et gérer les paiements. Voici comment cela est réalisé :

```
-- ============================
-- Fin d'une enchère et identification du gagnant
-- ============================
BEGIN;

-- Récupération de l'enchère gagnante
SELECT Email, PrixOffre, Quantite
FROM OFFRE
WHERE IdVente = :idVente
ORDER BY
    CASE
        WHEN (SELECT EstMontante
              FROM SALLEDEVENTE
              WHERE IdSalle = (SELECT IdSalle FROM VENTE WHERE IdVente = :idVente)) = 1
        THEN PrixOffre DESC
        ELSE PrixOffre ASC
    END
FETCH FIRST 1 ROWS ONLY;

COMMIT;
```

4.4 Mise à jour des stocks après une vente

Après la vente, il est vital de mettre à jour les stocks. Voici l'implémentation :

```
-- ============================
-- Mise à jour des stocks après la vente
-- ============================
BEGIN;

-- Récupération des détails du produit
SELECT Stock, IdProduit
FROM PRODUIT
WHERE IdProduit = :idProduit;

-- Mise à jour du stock du produit
UPDATE PRODUIT
SET Stock = Stock - :quantiteGagnante
WHERE IdProduit = :idProduit;

-- Vérification de la disponibilité du produit
```

```
UPDATE PRODUIT
SET DispoProduit = CASE
    WHEN Stock <= 0 THEN 0
    ELSE 1
END
WHERE IdProduit = :idProduit;

COMMIT;
```

Ces transactions SQL illustrent comment les fonctionnalités de notre système de gestion des enchères sont mises en œuvre.

5 Mode d'emploi du demonstrateur

Ce démonstrateur permet d'interagir avec une base de données via une interface en ligne de commande. Les utilisateurs peuvent effectuer diverses actions telles que consulter des utilisateurs, des salles de vente, des ventes, des catégories, des caractéristiques et des offres.

Vous pouvez exécuter le démonstrateur avec plusieurs options :

— **Initialiser la Base de Données avec des Données** :

```
make newrun
```

Cette commande réinitialise la base de données en remplissant les données à partir du dossier `data`.

— **Exécuter sans Réinitialisation** :

```
make run
```

Cela lance le programme sans réinitialiser la base de données.

Ensuite, vous aurez la possibilite d'entrer vos informations en tant qu'utilisateur(email, nom, prenom, adresse postale). Ainsi, vous pourriez faire ces differents choix :

1. **Afficher tous les utilisateurs** : Affiche une liste de tous les utilisateurs dans la base de données.
2. **Afficher les informations d'un utilisateur spécifique** : Permet de consulter les détails d'un utilisateur en particulier.
3. **Afficher les salles de vente** : Liste toutes les salles de vente.
4. **Afficher les salles de vente disponibles** : Montre uniquement les salles de vente non occupées.
5. **Afficher les ventes** : Affiche toutes les ventes en cours ou passées.
6. **Afficher les catégories** : Montre toutes les catégories de produits.

7. **Afficher les caractéristiques des produits** : Affiche les caractéristiques associées aux produits.
8. **Faire une offre** : Permet aux utilisateurs de soumettre une offre sur un produit.
9. **Voir les résultats des enchères** : Affiche les résultats d'une vente spécifique.

6 Bilan du projet

6.1 Retour d'Expérience

Le développement de ce projet a été une expérience enrichissante. Les principales leçons retenues incluent :

— **Importance de la planification :** La planification initiale a permis de structurer le développement et de définir des objectifs clairs magre les embuches. Nous avons essaye de remplir les differents objectifs que nous nous fixions.

— **Collaboration et Communication :** Le travail en équipe est essentiel. Malgré que nous ne nous connaissions pas personnellement, nous avons pu travailler ensemble en repartissant les taches. Des conflits d'idées ont eu lieu mais des réunions régulières ont permis de partager les avancées et de résoudre rapidement les problèmes.

— **Gestion des imprévus :** Plusieurs défis techniques ont été rencontrés, notamment des problèmes de connexion à la base de données et des erreurs de traitement des données. Ces défis nous ont appris beaucoup sur la conception des bases de donnees surtout le fait que l'interface machine doit etre en accord avec la base de donnees. Faire des aller-retour entre l'implantation de la base de données et l'implementation de l'application est important pour garantir une coherence des données.

6.2 Perspectives d'Amélioration

Plusieurs axes d'amélioration ont été identifiés :

— Amélioration du demonstrateur : Bien que fonctionnelle, l'interface pourrait être améliorée pour être plus intuitive et attrayante visuellement.

— Fonctionnalités Supplémentaires : limitation d'offres, un timer pour mieux determiner le flux des diverses offres...etc

— Apres la soutenance, Nous pensons qu'il est preferable pour le futur de faire des tables spécifiques pour chaque type de vente et les faire heriter de Vente de telle sorte a pouvoir avoir plusieurs types de vente dans une journee par exemple.