|  |  |
| --- | --- |
| Julien VILLETORTE | Licence Informatique |
| Projet de Base de données | |
| Année 2007-2008 | Professeur : Marinette Savonnet |

Sommaire

1. Introduction
2. Diagrammes
   1. Diagramme Use Case
   2. Diagramme de classes
3. Modèle de données
4. Création des tables sous Oracle
5. Exemple d’utilisation simpliste
6. Conclusion

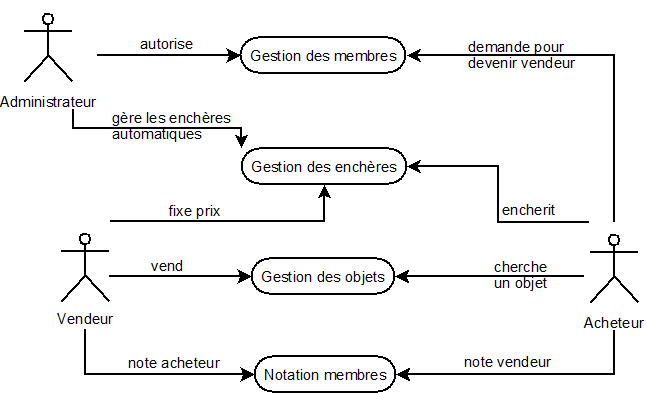
# Introduction

Ebay est à l’heure actuelle le site web de référence en matière de ventes aux enchères en ligne. Même si quelques concurrents existent, c’est Ebay qui a posé les bases du marché en ligne actuel, et son fonctionnement est bien rodé. C’est en grande partie la façon dont sont effectuées les enchères qui font sa force.

C’est pourquoi, nous allons ici essayer de reproduire un fonctionnement minimaliste du mécanisme d’Ebay, en s’appuyant sur des exemples de démarches de ventes, d’inscriptions, etc.

# Diagrammes

## Diagramme Use Case

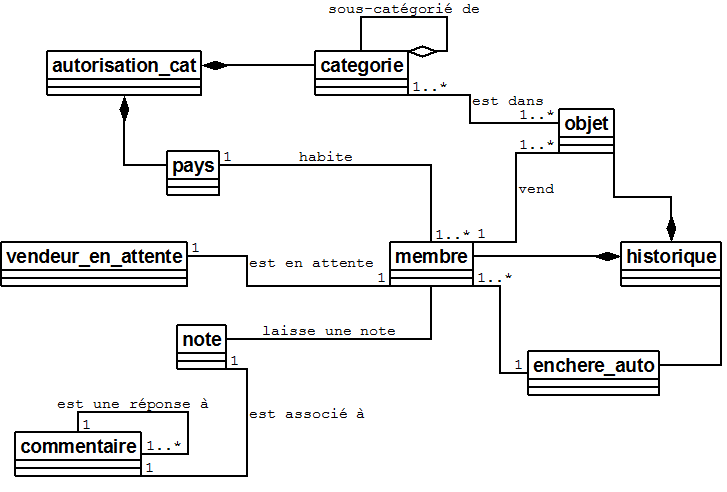


Un membre inscrit, est de base acheteur. S’il veut devenir vendeur, il doit faire une demande en faisant vérifier son identité auprès des administrateurs du site. L’acheteur cherche tout d’abord un objet dans les différentes catégories qui l’intéresse pour trouver ce qu’il souhaite. Une fois l’objet désiré trouvé, il enchérit dessus. Au préalable, cet objet a été déposé à la vente par un vendeur. Ce dernier en a fixé le prix minimum (et/ou autres prix : prix de réserve, etc.). Les éventuelles enchères automatiques que peut faire un acheteur sont gérées par les administrateurs d’Ebay. Au fur et à mesure que la transaction progresse, ces derniers mettent à jour les informations sur les différentes enchères des acheteurs.

Lorsque que la période d’enchère arrive à expiration, soit personne ne remporte l’enchère (prix de réserve non atteint, etc.), soit, comme dans la grande majorité des cas, l’enchère est remportée par un acheteur lambda. Dans ce cas, l’acheteur se doit d’envoyer le payement, avant que le vendeur n’envoie l’objet concerné (si celui-ci offrait la possibilité de livraison).

Une fois la transaction terminée (quelle qu’en soit l’issue), l’acheteur note le vendeur, et vice-versa. C’est ce qui permettra aux futurs vendeurs/acheteurs confronté à un membre, de savoir quel niveau de confiance lui accordé, et ainsi, mieux dirigé les recherches des acheteurs.

## Diagramme de classes



### 

### Explications :

|  |  |
| --- | --- |
|  | La classe Pays contient la liste de tous les pays, ainsi que l’âge légal de la majorité dans le pays en question. |
| La classe Categorie contient les différentes catégories dans lesquelles peuvent se trouver les objets en vente (cf. opérations). Une catégorie peut être la sous-catégorie d’une autre, l’attribut id\_parent permet donc de désigner la catégorie mère. Le nom et la description serviront à l’utilisateur pour savoir quels types d’objets se trouvent dans telle ou telle catégorie. |  |
|  | La classe Autorisation\_categorie est une classe association entre Categorie et Pays, elle permet de déterminer si une catégorie est autorisée ou non dans un pays. L’identifiant est donc : id\_cat,id\_pays |
| Un membre est caractérisé dans un premier temps par toutes les informations concernant son identité. Un numéro de membre lui est attribué quand il s’inscrit. Deux booléens, précisent si le membre a le droit d’acheteur et de vendre des objets (si il en a reçu l’autorisation). Un champ pour le numéro de carte bleue est aussi présent pour les vendeurs. Les deux derniers attributs, concernent les notes (moyennes) qui sont attribuées au membre, suivant les cas où celui-ci est vendeur ou acheteur. |  |
|  | Un objet est vendu par un membre id\_vendeur. Celui-ci fixe les différents prix possibles, s’il veut un prix de réserve, un prix d’achat immédiat, etc. Il donne une description détaillée de l’objet, et en donne l’état. Pour plus de facilité, l’identifiant du membre qui propose la plus haute enchère, est enregistré, ainsi que le montant de celle-ci. |
| Cette classe contient tous les codes de confirmation et les identifiants des membres en attente de validation pour devenir vendeurs. L’attribut date\_expiration, concerne la date jusqu’à laquelle le membre peut confirmer son code. Passé ce délai, il devra faire une nouvelle demande. |  |
|  | Cette classe est faite pour les différents commentaires qui peuvent être fait sur par rapport à une transaction. Si l’attribut id\_parent est renseigné, c’est que le commentaire en question est une réponse au commentaire dont l’identifiant est contenu dans id\_parent. |
| Une note est généralement donnée à un vendeur par l’acheteur, et vice-versa. Celui-ci est accompagné d’un commentaire. L’identifiant du membre qui met la note et celui de l’objet permettent de savoir quel rôle a joué le membre qui dépose la note, lors de la transaction. |  |
|  | La classe Historique permet de pouvoir afficher les différentes enchères sur un objet, même après la fin de la transaction, mais aussi, grâce au montant, d’obtenir l’identifiant de l’enchérisseur actuel (c’est-à-dire celui qui propose la plus grosse somme). |
| La classe Enchere\_auto, sert dans le cas particulier où un acheteur a demandé une enchère automatique sur un objet, à hauteur d’un montant maximum. A chaque fois que le montant évolue, la nouvelle enchère est renseignée directement dans l’historique. |  |

# Modèle de données

Nous pouvons traduire nos classes avec les relations suivantes :

Membre(id\_mbr, is\_acheteur, is\_vendeur, nom, prenom, dt\_naiss, adresse, Zip\_code, ville, téléphone, pays, login, mot\_de\_passe, email, num\_cb, note\_vendeur, note\_acheteur)

Pays(id\_pays, majorite\_pays)

Vendeur\_en\_attente(code, id\_mbr, date\_expiration)

Objet(id\_obj, id\_vendeur, id\_cat, frais\_envoi, description, etat, prix\_dep, prix\_reserve, prix\_achat\_immediat, assurance\_livraison, montant\_acquisition, id\_encherisseur\_actuel, mini\_surenchere, date\_fin)

Categorie(id\_cat, id\_parent, nom\_cat, description)

Autorisation\_cat(id\_cat, id\_pays, autorisee)

Historique(id\_ench, id\_obj, id\_mbr, montant, date)

Enchere\_auto(id\_ench\_auto, id\_obj, id\_mbr, ench\_max, date)

Note(id\_note, id\_obj, id\_mbr, id\_commentaire, note)

Commentaire(id\_commentaire, id\_parent, id\_mbr, commentaire, date)

# Création des tables sous Oracle

Voici les ordres SQL permettant de créer nos tables, en respectant les contraintes d’intégrité :

CREATE TABLE Pays  
(  
id\_pays integer PRIMARY KEY,   
majorite\_pays int(2) NOT NULL  
);

CREATE TABLE Membre  
(  
id\_mbr integer NOT NULL PRIMARY KEY,   
is\_acheteur bool,   
is\_vendeur bool,   
nom varchar(30) NOT NULL,   
prenom varchar(30) NOT NULL,   
dt\_naiss date NOT NULL,   
adresse varchar(60) NOT NULL,   
Zip\_code varchar(5) NOT NULL,   
ville varchar(15) NOT NULL,   
téléphone NOT NULL,   
pays NOT NULL REFERENCES Pays(id\_pays),   
login varchar(30) NOT NULL,   
mot\_de\_passe varchar(30) NOT NULL,   
email varchar(30) NOT NULL,   
num\_cb integer,   
note\_vendeur integer,   
note\_acheteur integer  
);

CREATE TABLE Vendeur\_en\_attente  
(  
code varchar(30) NULL PRIMARY KEY,   
id\_mbr NOT NULL REFERENCES Membre(id\_mbr),   
date\_expiration date  
);

CREATE TABLE Categorie  
(  
id\_cat NOT NULL PRIMARY KEY,   
id\_parent NOT NULL REFERENCES Categorie(id\_cat),  
nom\_cat varchar(30),   
description TEXT  
);

CREATE TABLE Objet  
(  
id\_obj NOT NULL PRIMARY KEY,   
id\_vendeur NOT NULL REFERENCES Membre(id\_mbr),   
id\_cat NOT NULL REFERENCES Categorie(id\_cat),   
frais\_envoi integer,   
description TEXT,  
etat ENUM('neuf', 'bon', 'moyen', 'mauvais'),   
prix\_dep integer,   
prix\_reserve integer,   
prix\_achat\_immediat integer,   
assurance\_livraison bool,   
montant\_acquisition integer,   
id\_encherisseur\_actuel NOT NULL REFERENCES Membre(id\_mbr),  
mini\_surenchere integer,  
date\_fin date  
);  
  
CREATE TABLE Autorisation\_cat  
(  
id\_cat integer NOT NULL REFERENCES Categorie(id\_cat),   
id\_pays integer NULL REFERENCES Pays(id\_pays),  
autorisee bool  
);

--ajout de la contrainte  
ALTER TABLE Autorisation\_cat ADD CONSTRAINT pk\_auto\_cat PRIMARY KEY(id\_cat, id\_pays);

CREATE TABLE Historique  
(  
id\_ench integer NOT NULL PRIMARY KEY,   
id\_obj integer NULL REFERENCES Objet(id\_obj),   
id\_mbr integer NULL REFERENCES Membre(id\_mbr),   
montant integer NOT NULL,   
date date  
);

CREATE TABLE Enchere\_auto  
(  
id\_ench\_auto integer NOT NULL PRIMARY KEY,   
id\_obj integer NULL REFERENCES Objet(id\_obj),   
id\_mbr integer NULL REFERENCES Membre(id\_mbr),   
ench\_max integer NOT NULL,   
date date  
);

CREATE TABLE Commentaire  
(  
id\_commentaire NOT NULL PRIMARY KEY,   
id\_parent integer NULL REFERENCES Commentaire(id\_commentaire),   
id\_mbr integer NULL REFERENCES Membre(id\_mbr),   
commentaire TEXT,   
date date  
);

CREATE TABLE Note  
(  
id\_note NULL PRIMARY KEY,   
id\_obj integer NULL REFERENCES Objet(id\_obj),   
id\_mbr integer NULL REFERENCES Membre(id\_mbr),   
id\_commentaire integer REFERENCES Commentaires(id\_commentaire),   
note integer NOT NULL  
);

# Exemple d’utilisation simpliste

Dans notre exemple, nous allons considérer les membres suivant, comme étant inscrit sur Ebay :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Id\_mbr** | **Login** | **…** | **Note\_vendeur** | **Note\_acheteur** |
| 9083 | Lala | … | 90 | 100 |
| 2130 | Truc | … | 86 | 94 |
| 1304 | Machin | … | 100 | 95 |

Le membre Lala veut vendre une montre aux enchères sur le site, il l’ajoute donc :

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Id\_obj** | **Id\_vendeur** | **Id\_cat** | **Frais\_envoi** | **Description** | **Etat** | **Prix\_dep** | **Prix\_reserve** |  |
| 67926 | 9083 | 68 | 5 | blabla | Bon | 80 | NULL |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Prix\_achat\_immédiat** | **Assurance\_livraison** | **Montant\_acquisition** | **Id\_encherisseur\_actuel** | **Mini\_surenchere** |
|  | 130 | False | 80 | NULL | 10 |

La requête suivante a été nécessaire pour ajouter cet objet :

INSERT INTO Objet(id\_obj, id\_vendeur, id\_cat, frais\_envoi, description, etat, prix\_dep, prix\_achat\_immediat, assurance\_livraison, montant\_acquisition, mini\_surenchere)   
Values(‘’, 9083, 68, 5, ‘blabla’, ‘bon’, 80, 130, false, 80, 10)) ;

Imaginons que Truc commence, en acceptant l’enchère de Lala :

UPDATE Objet   
SET id\_encherisseur\_actuel=2130  
WHERE id\_obj=67926 ;

Il faut aussi ajouter cette enchère dans l’historique :

INSERT INTO Historique(id\_obj, id\_mbr, montant, date)  
Values(67926, 2130, 80, sysdate);

Machin veut ensuite faire une enchère plus élevée, d’au minimum 10 euros, il faut donc mettre à jour la table Objet, et d’insérer une nouvelle valeur dans l’historique :

UPDATE Objet   
SET id\_encherisseur\_actuel=1304, montant\_acquisition=90  
WHERE id\_obj=67926 ;

INSERT INTO Historique(id\_obj, id\_mbr, montant, date)  
Values(67926, 1304, 90, sysdate);

La date de fin d’enchère est arrivée, et personne n’a renchérit, Machin remporte donc la montre de Lala.

Truc, étant frustré d’avoir raté l’enchère, veut consulter l’historique. La requête suivante est donc nécessaire :

SELECT m.login, h.montant, h.date  
FROM Membre m, Historique h  
WHERE m.id\_mbr=h.id\_mbr  
AND h.id\_obj=67926  
ORDER BY h.date DESC;

Ainsi, Truc peut constater que Machin a remporté l’enchère.

# Conclusion

Penser et concevoir une base de données pour un site tel qu’Ebay, demande beaucoup de réflexion et d’organisation. Celle-ci doit être pensée pour que l’utilisateur puisse avoir accès aux informations qu’il désire, le plus simplement et facilement possible. L’administration doit elle aussi pouvoir se faire sans difficulté.

La plus grande difficulté est de ne rien oublier. En effet, en simuler le fonctionner du site est le meilleur moyen de s’assurer que l’on n’oublie pas quelque chose de vitale dans notre modélisation. De plus il faut rendre la base évolutive, et ne pas la bloquer.