

Baie-électronique

Introduction

Le projet auquel vous allez participer a pour but de mettre en œuvre vos compétences en systèmes de gestion de bases de données relationnelles. De plus, vous vous essayerez à la programmation d'application utilisant une base de données (à travers des transactions). Le développement sera fait en Java en utilisant l'API JDBC.

Le projet est à faire en **équipes de quatre** et donnera lieu à une réunion de suivi intermédiaire, ainsi qu'à une soutenance en fin de projet. La constitution des équipes et la remise des livrables (rapport final et code source) se fera sur l'application *Teide*. L'évaluation se fera sur **les aspects bases de données du projet uniquement** en l'état au moment de la soutenance.

1. Description de l'application

La société *Baie-électronique* souhaite informatiser son service de ventes aux enchères. Elle vous propose donc de mettre en place la base de données nécessaire à cette application et de tester certaines fonctionnalités.

Les utilisateurs de *Baie-électronique* peuvent mettre en vente des produits ou enchérir sur des produits mis en vente par d'autres. Pour organiser un tant soit peu les ventes aux enchères, *Baie-électronique* se base sur le concept des Salles de Ventes. Une Salle de Vente propose la vente aux enchères d'une sélection de produits d'une même catégorie dans un même type d'enchères. Chaque vente, identifiée de façon unique, ne concerne qu'un seul produit et n'a lieu que dans une seule salle. Une catégorie de produit est identifiée par son nom et possède une description. Un produit est identifié de façon unique et est décrit par son nom, son prix de revient (prix à partir duquel le vendeur fait des bénéfices) et le stock proposé à la vente (un lot fait l'objet que d'une seule vente, éventuellement divisée en sous-lots). Un ensemble de caractéristiques (couples nom et valeur) permet de décrire spécifiquement le produit.

Les types de vente aux enchère sont variés mais ils possèdent tous un prix de départ à partir duquel démarreront les enchères. Une vente peut être montante (les offres sont toujours croissantes) ou descendante (le prix des produits diminue toutes les minutes, le premier utilisateur à faire une offre remporte la vente). Une vente, qu'elle soit montante ou descendante, peut autoriser un même utilisateur à enchérir plusieurs fois sur le même produit (en respectant le sens, bien sûr), ou seulement une seule fois. Une vente peut également être à durée limitée (la date et l'heure de fin sont alors précisées) ou à durée libre (le délai maximal entre deux offres est fixé à 10 minutes). Enfin, une vente peut être révocable, la vente est annulée si le prix de revient du produit n'est pas atteint, ou non révocable : le vendeur peut être amené à vendre à perte. Par défaut, les ventes sont montantes, non révocables, sans limite de temps et permettent à un même utilisateur d'enchérir plusieurs fois.

Une enchère, effectuée par un seul utilisateur, porte sur une vente en particulier et propose un prix d'achat. La date et l'heure de l'enchère sont également des éléments importants pour décider de sa validité et pour déterminer qui remporte la vente. De plus, une enchère peut ne concerner qu'un sous-lot du produit mis en vente, c'est pourquoi la quantité de produit concerné est toujours

précisée. Il est possible en pratique qu'une enchère se termine par la vente d'un (ou de quelques) sous-lots du produit concerné. L'objectif est alors que la vente rapporte le plus possible au vendeur, même si tous les produits ne sont pas vendus et si les sous-lots sont finalement vendus à des prix différents (en fonction des enchères qui ont été faites).

Les utilisateurs de l'application sont identifiés par leur email et décrits par leur nom, leur prénom et leur adresse postale (pour la livraison). Nous ne nous occupons pas ici des informations de paiement, cela étant géré par une autre application déjà en place.

2. Travail à réaliser

2.1. Modélisation du problème

La modélisation se décompose en deux temps. Dans un premier temps, vous aurez à **analyser le problème posé** pour en extraire les propriétés élémentaires, les dépendances fonctionnelles reliant ces propriétés, ainsi que tous les autres types de contraintes (contraintes de valeur, contraintes de multiplicité et autres contraintes). Cette analyse doit prendre en compte la brochure informatisée (schéma fourni). De cette analyse, vous devrez proposer ensuite un **schéma Entités/Associations** représentant les données nécessaires à l'application et leurs liens sémantiques (ce qui correspond à l'état cohérent de la base de données).

2.2. Implantation de la base de données

Vous devrez ensuite traduire le schéma Entités/Associations en un **schéma relationnel** que vous implanterez sur le SGBD Oracle disponible sur *ensioracle1*. Vous devrez insérer suffisamment de données pertinentes pour la suite du projet. **Vous préciserez également la forme normale de chacune des relations obtenues.**

2.3. Analyse des fonctionnalités

Vous devrez définir les requêtes SQL2 nécessaires pour réaliser les fonctionnalités suivantes en les regroupant en transactions, ce qui permettra d'assurer la cohérence globale de la base de données, même en cas d'accès concurrents :

- La mise en place d'une Salle de Vente et la sélection de produits déjà disponibles à la vente et permettant le choix du type d'enchères et du prix de départ ;
- L'enchère faite par un utilisateur sur un produit mis en vente dans une Salle de Vente ;
- Le processus de fin d'enchère déterminant le (ou les) utilisateur(s) ayant remporté une vente, en tenant compte du type d'enchère bien sûr.

Ces requêtes et transactions peuvent (doivent !) être testées sur Oracle (SQL*Plus) pour en vérifier leur bon fonctionnement, y compris pour des exécutions concurrentes.

2.4. Implantation des fonctionnalités

Les fonctionnalités précédemment étudiées devront être implantées en Java/JDBC. Vous pouvez choisir une interface texte ou graphique, cela n'a pas d'importance vu que **seuls les aspects Bases de Données seront évalués.**

3. Déroulement du projet

Le projet sera constitué de 15 heures en séances encadrées et 3 heures réservées pour les soutenances.

3.1. Séances encadrées

Les séances encadrées sont **obligatoires**. Lors de ces séances, les équipes devront avancer sur le projet et pourront poser des questions à leur encadrant. Attention : posez bien vos questions. L'encadrant jouera **soit le rôle du client** (et, dans ce cas, ne pourra traiter les aspects BD), **soit le**

rôle d'un expert en bases de données (mais, dans ce cas, il ne connaît rien à l'application). En fin de chaque séance, chaque équipe devra déposer une version actualisée de la documentation du projet **sans la valider** (validation uniquement en fin de projet). Ceci afin de pouvoir suivre le déroulement du projet et de pouvoir intervenir au plus tôt en cas de grosses erreurs/anomalies/difficultés.

3.2. Outils

Vous disposez de deux outils principaux pour le bon déroulement du projet :

- **Chamillo** : vous y trouverez les documentations techniques pour accéder à Oracle et pour utiliser JDBC, des liens Internet utiles, ainsi qu'un forum dans laquelle vous pourrez poser des questions importantes pouvant concerner toutes les équipes et y trouver des réponses (à consulter souvent, donc).
- **Teide** : l'application de gestion de projet. Vous devrez utiliser Teide pour constituer vos équipes, déposer vos rendus (documentation à chaque séance **sans la valider**, code source Java et SQL en fin de projet) et vous inscrire aux créneaux de soutenance.

3.3. Suivi

Le projet donnera lieu à une réunion de suivi intermédiaire entre l'encadrant et chacune des équipes. Cette réunion sera provoquée par les équipes elles-mêmes ou par l'encadrant en fonction de l'avancement du travail. Les thèmes discutés lors des suivis seront l'analyse et la modélisation Entités/Associations (~4^{ème} séance encadrée).

3.4. Livrables

- **Documentation du projet** : Vous devrez maintenir la documentation du projet tout au long de son déroulement. La documentation doit comprendre :
 - l'analyse du problème,
 - la conception Entités/Associations,
 - sa traduction en relationnel en précisant les formes normales des relations,
 - l'analyse des fonctionnalités (transactions), leur implantation sous forme de requêtes SQL2,
 - ainsi qu'un bilan du projet (organisation, points difficiles rencontrés, etc.)

Un petit mode d'emploi de votre application est également le bienvenu.

La documentation doit comprendre les explications nécessaires à sa compréhension et à la justification de vos choix.

- **Sources java et SQL2** : Vous devrez rendre en fin de projet un script SQL permettant de créer votre schéma relationnel, un script SQL permettant de peupler la base de données, l'implantation SQL des fonctionnalités, ainsi que le code source Java du démonstrateur.

3.5. Soutenance

Le projet se termine par une soutenance. Chaque équipe devra s'inscrire sur un créneau de soutenance via l'application *Teide*. La soutenance se compose de la façon suivante :

- 20 minutes pendant lesquelles vous devrez présenter votre projet (conception, implantation, fonctionnalités), faire une démonstration convaincante de votre prototype et faire un bilan du projet. L'objectif est de « vendre » le produit (montrer qu'il fait bien ce qui est demandé) et de « vendre » l'équipe (montrer que l'équipe a bien travaillé et que le produit est bien conçu).
- 10 minutes de questions pour le jury.

Les soutenances sont courtes, vous devez donc bien les préparer (scénario pour la démonstration, répétition, etc.)