

Projet Bases de Données

1 – 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 du démonstrateur sera fait en Java en utilisant l'API JDBC.

Le projet est à faire en **équipes de cinq** 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 uniquement sur les aspects bases de données du projet en l'état au moment de la soutenance.

2 – 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, acheter ou enchérir sur des produits mis en vente par d'autres (on rappelle qu'une enchère est une offre d'un prix supérieur à la mise à prix, ou au prix qu'un autre a déjà offert). Pour organiser un tant soit peu les ventes, *Baie-électronique* se base sur le concept des Salles de Ventes. Une Salle de Vente propose la vente d'une sélection de produits d'une même catégorie dans un même type d'offres. 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 ne fait l'objet que d'une seule vente, éventuellement divisée en sous-lots). Un ensemble de caractéristiques (couples regroupant le nom de la caractéristique et la valeur associée) permet de décrire spécifiquement le produit.

Les types de vente sont variés mais ils possèdent tous un prix de départ à partir duquel démarreront les offres. 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 faire plusieurs offres sur le même produit (en respectant le sens, bien sûr), ou limiter les utilisateurs à une seule offre par produit. 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 offre, effectuée par un seul utilisateur, porte sur une vente en particulier et propose un prix d'achat. La date et l'heure de dépôt de l'offre sont également des éléments importants pour décider de sa validité et pour déterminer qui remporte la vente. De plus, une offre 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.

3 – Travail à réaliser (15h encadrées + travail personnel, le tout x5)

Le travail se découpe en quatre étapes, décrites ci-dessous, dont seules les deux premières sont obligatoirement séquentielles. L'étape 3 peut être découpée et parallélisée, et l'étape 4 peut être commencée dès le début du projet.

3.1 – 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 contraintes contextuelles).

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).

Le schéma Entités/Associations doit être totalement justifié par l'analyse. Expliquez vos choix de modélisation dans la documentation du projet.

3.2 – Implantation de la base de données

Vous devrez ensuite traduire le schéma Entités/Associations en un **schéma relationnel** décrivant les noms des relations obtenues, les noms et types de leurs attributs, ainsi que les contraintes à vérifier pour chacune des relations. **Vous devrez également justifier vos choix de traduction et expliquer les points difficiles.** Précisez et justifiez la **forme normale de chacune des relations obtenues**.

Vous implanterez ensuite ce schéma relationnel sur le **SGBD Oracle disponible sur le serveur oracle1**. Vous devrez insérer suffisamment de données pertinentes pour la suite du projet.

3.3 – 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'enchères bien sûr.

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

Ces **requêtes et transactions doivent être testées** sur Oracle (**avec SQL*Plus ou SQL developer**, Adminer ne gérant pas les transactions) pour en vérifier leur bon fonctionnement, **y compris pour des exécutions concurrentes**.

3.4 - Implantation des fonctionnalités

Les fonctionnalités précédemment étudiées devront être implantées dans un **démonstrateur programmé en Java/JDBC**. Nous vous recommandons l'implantation d'une **interface textuelle simple** (quelques menus et saisies utilisateurs/affichages).

L'évaluation ne portera que sur les aspects Base de Données et pas sur le code Java.

4– Déroulement du projet

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

4.1– Séance encadrées

Comme toute épreuve pratique à l'Ensimag, **la présence aux séances encadrées est obligatoire et sera contrôlée**.

La première séance comprendra une présentation du projet, de JDBC et de l'analyse de fonctionnalités. Le travail se fera ensuite par équipe (créées préalablement via Teide). Vous pourrez poser des questions à votre encadrant à tout moment lors des séances, mais il jouera deux rôles alternativement : il sera **soit votre client** (qui connaît les besoins applicatifs, mais ne connaît rien en base de données), **soit un expert en bases de données** (qui ne connaît pas et ne veut pas connaître les besoins applicatifs).

Posez bien vos questions.

4.2- Outils

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

Chamillo : vous y trouverez les documentations techniques pour accéder à Oracle et pour utiliser et vous procurer JDBC (exemple de code, documentation, etc.), des liens Internet utiles, ainsi qu'un forum dans laquelle vous trouverez les réponses aux questions fréquentes (à consulter souvent, donc).

Teide : l'application de gestion de projet. Vous devrez utiliser Teide pour constituer vos équipes, déposer vos rendus (documentation du projet à déposer à chaque séance sans la valider, code source Java et SQL2, et supports pour la soutenance en fin de projet).

4.3- Suivi

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

4.4- Livrables

Vous aurez trois livrables à fournir pour ce projet :

1. **Documentation du projet** (un fichier PDF): Vous devrez maintenir la documentation du projet tout au long de son déroulement. La documentation doit comprendre :

L'analyse du problème sous forme de contraintes (DF, de valeur, de multiplicité, contextuelles). Expliquez vos choix sur les points difficiles.

La conception Entités/Associations noté en UML, comme vu en cours, en expliquant vos choix. Précisez bien les contraintes non représentées dans le schéma. Le schéma peut être réalisé avec l'outil Dia (disponible sur toutes les plateformes), n'importe quel outil de dessin vectoriel, voire même sur papier puis scanné, tant qu'il est lisible et respecte les notations graphiques vues en cours.

Sa traduction en relationnel en précisant les formes normales des relations et en justifiant vos choix. Les contraintes non implantables en relationnel devront également être listées.

L'analyse des fonctionnalités (transactions), leur implantation sous forme de requêtes SQL2, le tout bien commenté.

Un bilan du projet (organisation, points difficiles rencontrés, etc. en prenant du recul sur le travail effectué).

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

La qualité et la complétude de la documentation seront prises en compte dans la notation.

2. **Sources Java et SQL2** (un fichier ZIP ou TGZ) : Vous devrez rendre en fin de projet un script SQL2 permettant de créer votre schéma relationnel, un script SQL2 permettant de peupler la base de données, l'implantation des fonctionnalités (squelette des transactions et requêtes SQL2), ainsi que le code source Java du démonstrateur.
3. **Supports pour la soutenance** : Les slides prévus pour la soutenance devront être déposés sur Teide (visez ~15 minutes).

4.5 – Soutenance finale

La soutenance **durera en tout 25-30 minutes** divisées de la façon suivante :

~15 minutes pour que vous présentiez votre projet et faire une démonstration bien préparée (scénario, jeux de données en place, etc.) ;

~10 minutes pour que le jury puisse vous faire un retour sur le projet et vous poser des questions.

L'objectif de la soutenance est double : vous devez **convaincre le client que le produit répond bien à ses besoins**, et vous devrez **convaincre l'expert que votre équipe a fait du bon travail**. La soutenance doit donc reprendre les différents points de la documentation en insistant sur les points que vous jugerez les plus pertinents.

5 – Conclusion

Le projet se déroule en quatre semaines, soutenances incluses. **Cela va aller très vite. Une bonne organisation de l'équipe est primordiale.**

Il est recommandé de diviser les tâches dès le début du projet et de toujours garder les « candides » par tâche (des personnes qui ne participent pas du tout à cette tâche), de façon à ce qu'ils puissent apporter un regard critique sur ce qui a été fait lors des réunions d'équipe.