

VEHLIB pour tous

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 essaieriez 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 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 sur **les aspects bases de données du projet uniquement** en l'état au moment de la soutenance.

1. Description de l'application

Vous devez réaliser une base de données permettant la gestion d'un parc de véhicules en libre-service, sur le principe du Vélib (<https://fr.wikipedia.org/wiki/V%C3%A9lib%27>). Vous devrez gérer les véhicules, les stations, les abonnés, les forfaits et les locations. On vous demande également de réaliser un petit démonstrateur en Java/JDBC mettant en places les fonctionnalités décrites en section 2.3.

Les véhicules à gérer sont de diverses catégories, ils peuvent être des voitures électriques, des vélos, des vélos électriques, des vélos avec remorque, ou des petits utilitaires. Dans différents endroits de la ville se trouvent des stations (identifiées par leur nom) où les abonnés peuvent louer un véhicule d'une certaine catégorie. Chaque station, localisée par son adresse, offre un certain nombre de places pour les catégories de véhicules qu'elle peut accueillir.

Pour chaque catégorie de véhicule, on précise la durée maximale d'utilisation et le prix horaire de location, ainsi que le montant de la caution qui sera prélevée si la durée maximale est dépassée.

Chaque véhicule a un identifiant unique, appartient à une catégorie, et offre un certain nombre de places.

On souhaite garder une trace de tous les locations des véhicules: abonnés, dates, heures et stations de départ et d'arrivée. On souhaite également connaître à tout moment la station où est garé n'importe quel véhicule qui ne soit pas en location.

Les abonnés, identifiés par leur numéro de carte bancaire et décrits par leur nom, prénom, date de naissance et adresse postale, ont une réduction de 25% sur le tarif plein pour chaque forfait et de chaque location s'ils ont moins de 25 ans ou plus de 65 ans. Chaque abonné peut avoir un ou plusieurs forfaits (au plus un par catégorie de véhicule).

Un forfait, identifié de façon unique, concerne une seule catégorie de véhicule et un seul abonné. Il permet soit un nombre illimité de locations pendant une durée limitée (jour, mois

ou année), soit un nombre de locations limité pour une période illimitée. Les forfaits à durée limitée sont décrits par leur durée, la date de début de validité, un prix et une remise fonction de la durée, appliquée aux locations. Chaque utilisation d'un véhicule est gratuite pendant la première heure, puis s'applique un tarif horaire spécifique à la catégorie de véhicule concerné. Les forfaits à locations limitées sont décrits par leur prix et le nombre maximum de locations (gratuites).

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 concepts, les dépendances fonctionnelles reliant ces concepts, ainsi que tous les autres types de contraintes (contraintes de valeur, contraintes de multiplicité et autres contraintes). 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 implémenter le schéma Entités/Associations en un **schéma relationnel** sur le SGBD Oracle 12c disponible sur *ensioracle1*. Vous devrez insérer suffisamment de données pertinentes pour la suite du projet. Vous préciserez également la **forme normale** 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 :

- Facturation d'une location.
- Temps moyen d'utilisation par véhicules par mois.
- Temps moyen d'utilisation par catégorie de véhicule par mois.
- Catégorie de véhicule la plus utilisée par tranche d'âge de 10 ans.
- Taux d'occupation des stations sur la journée.

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 trouverez les réponses aux questions importantes pouvant concerner toutes les équipes (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, 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 (~3ème 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.
- 10 minutes de questions pour le jury.

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