

Agence de Voyages AWHY

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 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

L'agence de voyages AWHY désire informatiser son activité « réservation de voyages » en utilisant un SGBD relationnel. L'objectif est de développer pour l'agence une base de données permettant de simuler et de réserver un voyage. Dans ce but, l'agence a convenu avec ses partenaires (groupes hôteliers, offices de tourisme, etc.) d'une spécification de brochure informatisée, sous forme de relations, regroupant les informations sur des villes, des circuits, des monuments, des musées, des hôtels, etc.

Cette spécification devra être complétée par les relations autorisant un client potentiel de l'agence à réaliser une simulation de voyage et à venir ensuite valider (ou modifier en cas de non disponibilité) les choix de sa simulation en se rendant à l'agence.

Vous devrez réaliser un démonstrateur implémentant certaines des fonctionnalités de l'application finale.

1.1. Définition du schéma de la base de données

Les informations de l'application seront gérées par une base de données relationnelle. Une partie du schéma relationnel de cette base vous est fournie. Cette partie correspond à la spécification convenue avec les partenaires de l'agence. Vous devez compléter ce schéma relationnel pour prendre en compte les informations additionnelles nécessaires à la mise en place de l'application. Vous devez créer toutes les relations du schéma complet : les relations respectant la spécification et celles additionnelles nécessaires à la mise en place de l'application.

Brochure informatisée

La partie de schéma relationnel fournie correspond au catalogue des voyages (brochure), conçu avec les partenaires de l'agence. Elle concerne les informations sur les possibilités de voyages offerts par l'agence AWHY. On trouve des relations décrivant les pays et villes considérés par l'agence ainsi que les lieux de visites (monuments, musées, etc.), les circuits et hôtels proposés dans les villes. Ce sont :

Ville (nomVille char(20), pays char(20))

Une ville possède un nom, elle appartient à un pays. En vue de simplification, on supposera qu'il n'y a pas deux villes de même nom dans un pays donné.

LieuAvisiter (nomLieu char(20), ville char(20), pays char(20), adresseLieu varchar(50), descriptifLieu varchar(100), prix int)

Un lieu (un site, un monument, un musée) à visiter possède un nom, un descriptif. Il se trouve à une certaine adresse dans une ville et un pays et il en coûte un certain prix pour le visiter. Pour certains lieux, le prix peut être égal à 0.

Circuit(idCircuit char(5), descriptif char(50), villeDépart char(20), paysDépart char(20), villeArrivée char(20), paysArrivée char(20), nbJoursTotal int, prixCircuit int)

Un circuit possède un identifiant et un descriptif. Il est réalisé entre une ville de départ et une ville d'arrivée, dure le temps nbJours et coûte le prix prix.

DateCircuit(idCircuit char(5), dateDépartCircuit date, nbPersonnes int)

Un même circuit peut être proposé à des dates différentes (dateDépart) pour un certain nombre de personnes (à disposition de l'agence).

Etapes(idCircuit char(5), ordre int, nomLieu char(20), ville char(20), pays char(20), nbJours int)

Un circuit correspond à une liste de lieux à visiter (étapes) dans un certain ordre. Chaque étape référence un lieu à visiter sur un nombre de jours donné.

Hotel (nomHotel char(20), ville char(20), pays char(20), adresseHotel char(50), nbChambresTotal int, prixChambre int, prixPetitDejeuner int)

Un hôtel est situé à une certaine adresse dans une ville et dispose de nbChambresTotal chambres (pour l'agence), à un prix prixChambre. Il propose un petit-déjeuner à un prix prixPetitDejeuner.

Ces relations **ne doivent pas être modifiées** afin d'assurer la compatibilité avec les applications des partenaires.

Schéma à définir

Avec les informations dont elle dispose, l'agence AWHY veut proposer à ses clients des simulations de voyage. Une simulation inclut :

- Le nom de la personne (client déjà connu ou potentiel) demandant la simulation,
- les dates de départ et d'arrivée envisagées pour le voyage,
- le nombre de personnes participant au voyage,
- les différentes options de réservation prises par le client lors de l'examen des informations proposées, c'est-à-dire le pays envisagé pour le voyage, les villes visitées, les lieux, circuits et hôtels sélectionnés,
- le coût total du voyage (calculé).

Suite à une simulation de voyage validée, l'utilisateur obtient un numéro de dossier (de simulation) avec lequel il peut se présenter à l'agence pour réaliser la réservation du voyage. Il faut alors valider les informations obtenues lors de la simulation : vérifier que les dates et les options de réservation sont toujours valides (nombre de places disponibles dans les circuits et nb de chambres disponibles dans les hôtels suffisants, en supposant une chambre par personne), réaliser les réservations auprès des organismes concernés et calculer le coût final du voyage.

Pour cela un dossier de réservation est créé pour le client. On devra également créer le client dans la base si celui-ci n'existe pas déjà. Un client a identifiant unique, un nom, un type (société, groupe, individuel) et une année d'enregistrement auprès de l'agence. Sur l'ensemble des clients, il ne peut pas exister de client sans réservation.

Pour compléter le schéma donné précédemment vous devez donc définir les relations nécessaires au stockage des informations de simulations, des réservations et des clients.

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 :

- **Consultation de la brochure**
Pourvoir consulter toutes les offres de l'agence.
- **Simulation d'un voyage**
Incluant l'éventuelle saisie du nom du client (s'il n'est pas connu), la sélection des éléments dans la brochure, le calcul du coût du voyage, et la validation de la réservation.
- **Réservation d'un voyage**
Incluant la vérification des disponibilités des éléments de la simulation et les modifications nécessaires (via la brochure), l'éventuelle saisie des informations du client (nom, prénom, adresse, type de paiement, etc.), le calcul du coût final du voyage et la validation de la réservation.

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. Dérroulement 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 (~4ème 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.)