

Festival du Cirque

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

Les organisateurs d'un prestigieux festival de cirque, qui se déroule une fois par an, demandent que vous leur construisiez une application exploitant une base de données pour gérer les inscriptions des artistes participants, la sélection des numéros et l'organisation des différents spectacles.

Ce festival de cirque dure quatre jours et chaque demi-journée est consacrée à un spectacle qui regroupe des numéros portant sur le même thème (magie, acrobaties, dressage d'animaux, ...). Les numéros présentés au festival sont une sélection de numéros présentés par différents cirques (appelés *cirques participants*). Les numéros candidats durent 10 à 30 minutes, ils sont présentés par une ou plusieurs personnes (appelées *artistes*) qui appartiennent à l'un des cirques participants. Un jury prépare le festival et fait la sélection : il regroupe une trentaine d'*experts* (eux-mêmes artistes), ayant chacun une ou plusieurs spécialités correspondant chacune à l'un des thèmes. Chaque numéro est évalué par cinq experts, dont trois spécialistes du thème concerné et deux non spécialistes. Chaque expert donne une *évaluation* (petit commentaire) et met une *note* comprise entre 0 et 10. Les experts ne doivent pas proposer eux-mêmes de numéros ni être du même cirque participant que les artistes des numéros qu'ils évaluent. Chaque expert ne peut expertiser plus de 15 numéros. A partir des notes, le jury classe les numéros, choisit les meilleurs et les affecte aux différents spectacles.

Les organisateurs ont besoin des informations suivantes pour la préparation du prochain festival. Chaque numéro candidat est caractérisé par un code unique, son titre, son thème, son résumé et sa durée en minutes. On retient aussi le nombre d'artistes présents sur scène avec mention de l'artiste principal à qui envoyer la réponse d'acceptation ou de refus, les cinq experts avec les évaluations et notes qu'ils mettront au numéro. Si le numéro est accepté, le spectacle auquel il sera présenté est également mémorisé.

Chaque artiste possède un identifiant unique, son nom, son prénom, sa date de naissance, son (ou ses) pseudonyme(s) s'il en a, son cirque d'origine (cirque participant), sa ou ses spécialités, son numéro de téléphone, le ou les numéros auxquels il participe, puisque les artistes sont susceptibles de présenter plusieurs numéros. Pour chaque expert, on note son nom, son prénom, sa date de naissance, son (ses) pseudonyme(s) éventuel(s), son cirque d'origine, sa ou ses spécialités, son numéro de téléphone et les numéros qu'il évalue avec l'évaluation et la note qu'il leur attribue.

Pour chaque spectacle, on garde un identifiant unique, le thème, le jour, l'heure de début (9h ou 14h), le présentateur (l'artiste qui anime le spectacle et présente les numéros ; cela peut être un expert ou un artiste qui n'intervient pas dans ce spectacle), la liste des numéros du spectacle et le prix d'entrée pour le spectacle (tous les spectacles n'ont pas le même prix d'entrée).

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 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**.

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 *oracle1*. 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 :

- **Inscriptions** : L'application devra fournir aux organisateurs un moyen de saisir les experts, les artistes proposant un ou participant à des numéros, ainsi que les descriptions des numéros candidats.
- **Évaluation des numéros** : L'application devra permettre de désigner un jury de cinq experts pour chaque numéro, ainsi que de saisir leurs évaluations et notes. Les propositions de jurys doivent respecter les contraintes décrites précédemment.
- **Planification des spectacles** : En fin de séance d'audition, l'application devra permettre de consulter le classement des numéros par thème de façon à ce que l'ensemble du jury puisse sélectionner les meilleurs et les affecter à des spectacles.
L'application devra veiller à la bonne organisation des spectacles (numéros de même thème, pas plus de 180 minutes par spectacle, sélection du présentateur, etc.). Elle devra permettre d'éditer le programme de chaque spectacle.

Ces requêtes et transactions peuvent (doivent !) être testées sur Oracle (SQL*Plus, *Adminer* ne gérant pas les transactions) 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 textuelle 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

La présence aux séances encadrées est **obligatoires** et **sera contrôlée**. 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), jamais les deux à la fois.

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 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, et supports pour la soutenance 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 du suivi consèrnera 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.
- **Supports pour la soutenance** : Si vous utilisez des slides pour la soutenance, veuillez les déposer sur Teide.

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 à 25 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).
- 5 à 10 minutes de questions pour le jury.

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