

Projet Bases de Données et Conception Objet

1 – Introduction

Ce projet a pour but de mettre en œuvre vos compétences en systèmes de gestion de bases de données relationnelles et en conception objet. Il vient en application du cours de « Principes des systèmes de gestion de bases de données » (PSGBD) et du cours d'« Analyse, conception et validation des logiciels » (ACVL). Vous devez réaliser l'analyse, la conception et la programmation complète d'une application de gestion d'un festival de cirque en utilisant une base de données. Vous devez notamment concevoir et implanter un schéma relationnel. Le programme doit être écrit en *Java*, et les accès à la base de données effectués grâce l'API *JDBC*.

Le projet est à faire en équipe de 5 élèves, il donnera lieu à un rendu intermédiaire, à la remise d'un rapport, ainsi qu'à une soutenance comprenant, entre autres, une démonstration. L'évaluation concerne l'analyse, la conception et l'implémentation, à la fois sur les aspects bases de données et sur le travail de conception orientée objets.

2 – 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 peuvent être des créations présentées pour la première fois ou des numéros déjà existant. Ils durent 20 à 40 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 spécialité correspondant 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 évaluations et 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 proposé est caractérisé par un code unique, son titre, son thème, son résumé et sa durée moyenne en minutes. On maintient 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 ont mises au numéro. Si le numéro est accepté, le spectacle et l'heure à laquelle il sera présenté sont également mémorisés.

Chaque artiste possède un identifiant unique, son nom, son prénom, sa date de naissance, son (ou ses) pseudonyme(s), son cirque d'origine (cirque participant), son adresse, le ou les numéros auxquels il participe, puisque les artistes sont susceptibles de présenter plusieurs numéros. S'il est artiste principal, on enregistre en plus son numéro de téléphone. Pour chaque expert, on note son nom, son prénom, sa date de naissance, son (ses) pseudonyme(s) éventuel(s), son cirque d'origine, son adresse, 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, l'heure de fin, le présentateur (celui qui anime le spectacle, présente les artistes et les numéros ; cela peut être un expert ou un artiste de l'un des numéros du spectacle), la liste des numéros du spectacle, avec leur heure de passage et le prix d'entrée pour le spectacle (tous les spectacles n'ont pas le même prix d'entrée).

Inscriptions

L'application devra fournir aux organisateurs un moyen de saisir les experts, les artistes proposant un ou plusieurs numéros, ainsi que les descriptions des numéros candidats.

Évaluation des numéros

L'application devra proposer automatiquement un jury de cinq experts pour chaque numéro au fur et à mesure de la saisie de ces derniers. Les propositions de jurys doivent respecter les contraintes décrites précédemment. En dernier recours, l'application devra également permettre de saisir des jurys « à la main » si aucune proposition correcte ne peut être générée automatiquement.

L'évaluation des numéros n'est pas complètement gérée par le festival. Chaque expert peut donner son évaluation quand il le souhaite. Pour les numéros n'ayant pas pu être évalués complètement, et en particulier pour les créations, l'application devra proposer, sur les deux jours précédant le festival, une planification de ces numéros, avec convocation du jury correspondant.

Les experts devront pouvoir saisir directement leurs évaluations et notes dans l'application après chaque audition.

Planification des spectacles

En fin de séance d'audition, l'application devra fournir le classement des numéros par thème de façon à ce que l'ensemble du jury puisse sélectionner les meilleurs et les affecter dans des spectacles.

L'application devra veiller à la bonne organisation des spectacles (numéros de même thème, pas plus de 3 heures par spectacle, sélection du présentateur, etc.). Elle devra permettre d'éditer la liste des spectacles et pour chaque spectacle, le programme.

Présentation de l'application

L'application doit fournir les fonctionnalités suivantes :

- permettre à un expert de sélectionner un numéro et de saisir rapidement son évaluation ;

- afficher le programme global du festival, et pour chaque spectacle afficher son programme ;

Cette application doit fournir le choix entre 2 interfaces : une en ligne de commande et une interface graphique. L'interface graphique peut être activée via un paramètre de la ligne de commande. Durant le projet, il sera mis la priorité sur l'interface en ligne de commande.

Les données nécessaires au bon fonctionnement de l'application pourront être chargées directement dans la base de données. Il est également vivement recommandé de créer des scripts SQL pour créer les tables, les peupler, et les resetter (jeu de tests initial), afin de simplifier le développement et la création de scénarii pour la maintenance.

3 – Travail à réaliser (9h encadrées + travail personnel)

Les différentes tâches à réaliser sont présentées dans le schéma de la page 6. Chaque rectangle du schéma correspond à une tâche à réaliser par une sous-équipe. Hormis les dates de rendu (intermédiaire et final), le temps à passer sur chaque tâche est indicatif et la répartition des membres sur les tâches (combien de personnes et qui) est laissée à votre appréciation.

Liste des différents points à aborder :

- 1) **Analyse** : analyse fonctionnelle, analyse statique
- 2) **Conception de la base de données** : élaboration du schéma conceptuel, conception de la base de données, analyse des accès à la base de données
- 3) **Conception de l'application** : architecture de l'application, conception objets
- 4) **Implémentation** : instanciation de la base de données, programmation de l'application.

Réaliser le projet revient à répondre à l'ensemble de ces points (ils sont détaillés dans la suite de l'énoncé). Pourtant, cette liste ne doit pas être prise chronologiquement. C'est une structuration logique du travail, qui pourra être reprise dans la documentation finale. Nous vous proposons de réaliser l'ensemble de ces points avec la répartition (thématique et chronologique) suivante.

Première semaine

Nous vous conseillons de vous répartir en deux sous-équipes, pour réaliser les deux tâches suivantes :

Tâche A : analyse statique et schéma conceptuel de la base de données

- a) **Analyse statique**. Inventorier les données élémentaires, identifier les dépendances fonctionnelles et les contraintes (invariants) entre les données persistantes.
- b) **Élaboration du schéma conceptuel**. Construire, à partir de l'analyse statique un schéma Entités/Associations (ce schéma correspond à un diagramme de classe d'analyse). Expliquer ce schéma (notamment les points difficiles de sa construction). Préciser les contraintes d'intégrité non représentées dans le schéma Entités/ Associations.

Tâche B : analyse fonctionnelle et architecture de l'application

- a) **Analyse fonctionnelle**. Identifier les acteurs et les cas d'utilisation, à une granularité très grosse et faire un diagramme de cas d'utilisation. Préciser le fonctionnement global du système (en vous aidant si nécessaire d'un ou plusieurs diagrammes états/transitions pertinents). *Le but de cette analyse est principalement de faire en sorte que l'ensemble de l'équipe ait la même lecture du cahier des charges.*

- b) **Architecture de l'application.** Proposer une architecture pour l'application. Préciser en particulier l'architecture permettant d'intégrer les interfaces utilisateur et la base de données à votre logiciel. Pour vous aider, vous pourrez considérer des scénarios typiques pour :
- une action de l'utilisateur sur l'interface.
 - le déclenchement et l'exécution d'une requête depuis l'application.

Ces scénarios pourront être illustrés par un diagramme de séquence ou de collaboration chacun.

Bien entendu, les sous équipes des tâches A et B devront communiquer très régulièrement car au début du projet, il y a beaucoup d'éléments à fixer !

Rendu intermédiaire : le 24 mai

Vous rendrez un résumé des résultats des tâches A et B. Précisément, nous vous demandons de rendre sur Teide :

- le schéma entité association de votre base de données, sous la forme d'un pdf, format A4
- le schéma ou les schémas de l'architecture de votre logiciel, sous la forme d'un pdf, format A4.

Deuxième semaine

Nous vous conseillons de vous répartir en trois sous-équipes, pour réaliser les trois tâches suivantes :

Tâche C : schéma relationnel et implémentation de la base de données

Traduire le schéma Entités/Associations en **relationnel** ; écrire les **tables et les contraintes**. Préciser et justifier la **forme normale** des relations, les **contraintes d'intégrité** implantées, ainsi que les **autres contraintes** ne pouvant être implantées en SQL2 (et qui sont donc à vérifier dans le code applicatif Java). **Instancier la base de données** ; elle doit être créée sur le serveur Oracle 12c disponible sur le serveur ensioracle1.

Tâche D : analyse des accès à la base de données

Identifier les requêtes nécessaires aux différentes fonctionnalités à l'aide de l'analyse fonctionnelle (que vous pourrez compléter si besoin) et les écrire. Délimiter les transactions permettant d'implanter les fonctionnalités tout en préservant la cohérence globale des données.

Tâche E : conception objet et implémentation du cœur de l'application

Faire un diagramme de classes logicielles et implémenter en Java les classes du cœur de l'application. Elles sont définies en priorité sur toutes les autres car le reste du logiciel va en dépendre.

Troisième semaine et fin

Nous vous conseillons de vous répartir en trois sous-équipes, pour réaliser les trois tâches suivantes :

Tâche F : interface graphique

Dessiner et programmer, en Java, l'interface graphique du logiciel. Cette étape requiert de savoir programmer une interface graphique en Java : attention à prévoir un temps d'apprentissage, si nécessaire dans votre organisation. Ne pas oublier d'intégrer et de tester régulièrement !

Tâche G : implémentation des accès à la base de données

Implémenter, en Java, le module applicatif des accès à la base de données. L'accès à la base de données se fera avec l'API JDBC : attention à prévoir un temps d'apprentissage dans votre organisation. Ne pas oublier d'intégrer et de tester régulièrement !

Tâche H : interface textuelle

Programmer, en Java, l'interface textuelle et les exécutables les utilisant. Ne pas oublier d'intégrer et de tester régulièrement !

4 – Documentation

Journal et bilan

Rédiger un **bilan** de ce projet (répartition des tâches, difficultés rencontrées, leçons retenues, *etc.*). Pour vous aider dans cette tâche, il vous est demandé de maintenir **un journal** de projet dans lequel vous préciserez (en quelques lignes) ce qui a été réalisé au cours de chaque séance encadrée de projet (ou en dehors de ces séances). Ce journal devra être déposé sur l'application *Teide* à la fin de chaque séance encadrée (déposé, mais non validé) et pourra être consulté à tout moment par vos encadrants tout au long du projet.

Documentation finale

Vous rendrez une documentation du travail réalisé pendant le projet. Elle devra contenir quatre parties correspondant à l'analyse, la conception et l'implantation de la base de données, la conception de l'application, et le bilan.

4 – Soutenance (Vendredi 2 juin 2017 à partir de 14h00)

Votre projet donne lieu à une soutenance de **30 minutes** (inscription sur les créneaux horaires *via* l'application *Teide*), suivie de 10 minutes de questions, durant laquelle vous devrez :

- présenter votre modélisation entités-associations et son implantation en relationnel
- présenter votre architecture
- présenter le diagramme de classes logicielles
- faire une démonstration des fonctionnalités ; prévoir un scénario montrant le bon fonctionnement des transactions
- faire un bilan du déroulement du projet

L'objectif est de vendre **votre produit** (en montrant qu'il fait bien ce qu'on attend) **et votre équipe de projet** (en montrant que le produit a été bien construit).

Note : Il est très fortement recommandé de consulter régulièrement les pages consacrées à ce projet.