Projet - Partie BD

1 Position du problème par incréments

Un directeur de théâtre désire informatiser son système de réservation de places pour les spectacles qui se déroulent dans son théâtre au cours d'une même saison. A la fin d'une saison, le contenu de la base de données est archivé et vidé. Pour les besoins du sujet les données de l'application décrites ci-après ont été simplifiées. Dans le cadre de réunions avec le directeur les différents incréments on été définis.

I1_LesSpectacles. Un spectacle est identifié par un numéro et on connaît son nom, son prix de base, le public cible (1-5 ans, jeune public, tout public ou adultes) et le type de spectacle (opéra, drame, humoristique, musical ou cirque). Sur les spectacles du type opéra, on indique s'il intègre une orchestre. Sur les spectacles du type humoristique on indique s'il s'agit d'un *OneWomen(Men)Show*. Un spectacle fait généralement l'objet de plusieurs représentations proposées à des moments différents. Il y a au plus une représentation par spectacle et par jour. Une représentation débute à un moment donné exprimé à la granularité de l'heure (par exemple 28/11/2007 20H).

I2_LesPlaces. La seule salle du théâtre est partitionnée en zones numérotées, regroupant chacune un ensemble de places. Une place est identifiée par un numéro de rang (unique par salle), et un numéro de place dans le rang. Une zone est associée à une seule catégorie tarifaire (orchestre, balcon, poulailler, etc). Une catégorie peut être associé à plusieurs zones. Toutes les places de la même zone sont dans la même catégorie. Un taux par rapport au prix de base est associé à chaque catégorie (ex. 1 pour le poulailler, 1.2 pour l'orchestre et 1.5 pour le balcon). Ce taux est fixé pour l'ensemble des spectacles.

I3_LesTickets. Chaque place vendue par représentation fait l'objet de l'émission d'un ticket identifié par un numéro de série et estampillé par la date au moment de l'emission (instant à la granularité de la seconde). Un achat concerne un ou plusieurs places (i.e., plusieurs tickets) se traduisant par la création d'un dossier achat (identifié par un numéro) auquel est associé le prix global des places. Un suivi du nombre de places disponibles pour chaque représentation programmée pour un spectacle est envisagé.

I4_LesRéductions. Pour certaines représentations, le prix des places fait l'objet d'une promotion. Ainsi, à chaque représentation est associé un taux de promotion (1 : pas de promotion, 0,5 : promotion de 50, etc.). Les personnes qui achètent des places peuvent être des spectateurs ordinaires ou des adhérents et bénéficier d'un tarif réduit. Certaines personnes (étudiants, scolaires, militaires et séniors) bénéficient aussi d'une réduction. Un taux de réduction est associé a chacun de ces types (1 pour le tarif sans réduction, 0.8 pour le tarif réduit de 20%, etc.). Le cas échéant, la réduction vient s'ajouter à la promotion associée à la représentation et éventuellement à d'autres promotions (ex. un adhérent avec 20% qui est aussi étudiant avec 10% de réduction, il aura 30% de reduction).

I5_LesUtilisateurs. L'application doit être mise en ligne afin de permettre à des utilisateurs identifiés d'acheter des places. Les utilisateurs sont identifiés par leur *login*. Ils sont décrits par leur nom, leur prénom, leur adresse électronique et leur mot de passe. Un utilisateur peut réserver des places avant de procéder à leur achat. Une place réservée ne peut être achetée que par l'utilisateur qui a

effectué la réservation. Une réservation pourrait être ensuite rendue disponible par le client qui l'a réservée.

I6_LesConfigurations. Selon les spectacles, toutes les places de certaines zones sont retirées de la vente : en effet, dans certaines mises en scène, une partie des tribunes est occupée par le spectacle. La configuration de la salle, c'est-à-dire l'ensemble des zones dont les places sont mises à la vente, est la même pour toutes les représentations d'un même spectacle.

Question 1 :
Fournir un diagramme de classes UML permettant de modéliser les données de cette application.

Question 2 :
Déduire un schéma relationnel du diagramme proposé à la question précédente. Préciser les relations, les vues, les domaines et les contraintes d'intégrité.

Question 3 :
Réaliser le code SQL DDL pour implémenter le schéma relationnel (*create tables*) en SQLite. Insérer au moins 5 n-uplets dans chacune des tables (*insert*). Créer un fichier des insertions produisant des erreurs, mettant en lumière les contraintes (comme le TP DDL-DML)

Question 4 :
Réaliser un TRIGGER permettant de contrôler une contrainte avancé. C'est-à-dire, qui ne peut pas être contrôlée par les contraintes de base (PK, FK, CHECK, NOT NULL, ...)