

Gestion d'un zoo

1. Contexte

Le directeur d'un Zoo a informatisé la gestion de son établissement. Dans ce Zoo, on trouve des animaux répertoriés par type (lion, léopard, girafe, escargot...). Chaque animal possède un nom (Charly, Arthur, Enzo...) qui l'identifie de façon unique, une date de naissance et un pays d'origine. On retient également les maladies que chaque animal a contractées depuis son arrivée au Zoo.

Les animaux sont logés dans des cages. Chaque cage peut recevoir un ou plusieurs animaux. Certaines, cages peuvent être inoccupées. Une cage correspond à une certaine fonctionnalité qui n'est pas forcément liée à un type d'animal donné. Une cage est identifiée par un numéro, elle est située dans une allée identifiée aussi par un numéro. Des animaux de type différent ne peuvent pas cohabiter dans une même cage.

Des personnes sont employées par le Zoo pour entretenir les cages et soigner les animaux. Chaque personne est identifiée par son nom, et on connaît la ville où elle réside. Les personnes sont affectées à un des deux postes, gardien ou responsable. Un gardien s'occupe d'une ou de plusieurs cages, et un responsable a la charge de toutes les cages d'une ou de plusieurs allées. Une allée est sous la responsabilité d'un seul employé et toute cage occupée par au moins un animal est gardée par au moins un gardien ; une cage inoccupée peut être affectée à un et un seul gardien pour entretien.

Le directeur désire gérer de manière automatique les historiques des affectations des gardiens : avant d'enregistrer une affectation, une suppression ou une modification d'affectation d'un gardien à une cage, le système doit garder la trace de l'affectation avant la modification. La date de la modification est fournie par le système, au moment de la transaction.

L'application utilise une base de données relationnelles dont le schéma est donné ci-dessous. Les identifiants des relations sont les attributs notés en caractères soulignés.

LesAnimaux (nomA, sexe, type, pays, anNais, noCage)

$\{ \langle n, s, t, p, a, c \rangle \in \text{LesAnimaux} \iff \text{l'animal de nom } n, \text{ sexe } s \text{ et de type } t \text{ est originaire du pays } p. \text{ Son année de naissance est } a. \text{ Il est logé dans la cage de numéro } c. \}$

LesMaladies (nomA, nomM)

$\{ \langle a, m \rangle \in \text{LesMaladies} \iff \text{l'animal de nom } a \text{ a contracté au zoo, la maladie } m \}$

LesCages (noCage, fonction, noAllée)

$\{ \langle n, f, a \rangle \in \text{LesCages} \iff \text{la cage de numéro } n \text{ est de type } f, \text{ elle est située dans l'allée } a. \}$

LesEmployés (nomE, adresse)

$\{ \langle e, a \rangle \in \text{LesEmployés} \iff \text{l'employé de nom } e \text{ réside dans la ville } a. \}$

LesResponsables (noAllée, nomE)

$\{ \langle a, e \rangle \in \text{LesResponsables} \iff \text{l'allée de numéro } a \text{ est sous la responsabilité de l'employé de nom } e. \}$

LesGardiens (noCage, nomE)

$\{ \langle c, e \rangle \in \text{LesGardiens} \iff \text{l'employé de nom } e \text{ est chargé de l'entretien de la cage de numéro } c. \}$

La description des domaines est la suivante :

dom (adresse) = chaîne de caractères ("Noumea", "Papeete", "Sartene"...)

dom (anNais) = [2000, ∞[

dom (fonction) = chaîne de caractères ("grand aquarium", "fosse", "fauves"...)

dom (noAllée) = [1, .., 999]

dom (noCage) = [1, .., 999]

dom (nomA) = dom(nomE) = dom(nomM) = dom(pays) = chaîne de caractères

dom (sexe) = {"femelle", "male", "hermaphrodite"}

dom (type) = chaîne de caractères

Les contraintes d'intégrités référentielles sont :

LesGardiens[noCage] \subset LesCages[noCage] LesResponsables[nomE] \subset LesEmployés[nomE]

LesGardiens[nomE] \subset LesEmployés[nomE]

LesMaladies[nomA] \subset LesAnimaux[nomA] LesHistoiresAff[nomE] \subset LesEmployés[nomE]

2. Mise en place

Vous devez créer la base de données Zoo sous la console SQLPLUS d'Oracle à l'aide du script *zoo.sql*

Vous pourrez utiliser DBeaver (ou une console SQLPlus) pour vérifier le contenu de la base.

3. Application Java

Le travail consiste à mettre en œuvre quelques fonctionnalités dans une petite application Java utilisant l'API JDBC pour accéder à la base. Aucune IHM n'est demandée.

Le modèle de données est disponible dans le dossier *modele*.

Les DAO pour accéder à la base sont le dossier *DAO*.

Le programme principal est dans le dossier *main*.

Les classes utilitaires sont dans le dossier *util*.

Vous utiliserez NetBeans, Eclipse ou Visual Studio Code pour développer vos programmes. Un nouveau projet Java Application doit être créé.

Vous devez juste ajouter à votre projet l'API *ojdbc8.jar* comme jar externe après avoir copié les dossiers dans votre projet.

Implémenter les trois fonctionnalités suivantes dans *ZooApp* (vous manipulerez exclusivement des objets métiers issus du modèle et des appels DAO):

- **Affichage de la liste des animaux** avec le numéro de cage où chacun d'eux se trouve, et leur liste de maladies.
- **Ajouter un nouvel animal** en demandant les caractéristiques de l'animal à l'utilisateur. Lister l'ensemble des cages compatibles. L'utilisateur choisit parmi ces dernières la cage où l'animal sera logé. L'utilisateur pourra également saisir une liste de maladies pour cet animal.
- **Affecter un gardien à une cage.** On fait l'hypothèse que le gardien existe déjà. Le nom du gardien est donné par l'utilisateur. Les cages compatibles avec les spécialités du gardien sont listées, puis l'utilisateur choisit une cage parmi cette liste.

L'implémentation devra suivre un modèle DAO. Vous aurez donc à développer les DAO pour manipuler de manière transparente ce modèle de données. Cependant, vous implémenterez uniquement les DAO utiles à vos fonctionnalités et les méthodes utiles de chacun des DAO. Les DAO sont disponibles dans le dossier *DAO* (vous pouvez modifier ces modèles ou en ajouter au besoin mais cela ajoutera du travail).