Rapport de Projet de BDAv

Par Samuel ELBEZ 21200353 & Romain STASYSZYN 21305734

Introduction

Le but de ce projet est d'implémenter, en PostgreSQL, la gestion d'une compagnie de théâtre appelé *Le Théâtre du Chrysanthème*.

L'achat et la vente de spectacles ainsi que les dépenses occasionnées, les subventions et la gestion de la billetterie permettent de calculer et d'estimer la rentabilité du théâtre et de ces représentations.

La compagnie peut avoir des représentations, de ces créations ou des pièces qu'elle a acheté, en interne dans leur salle modulable ou bien en externe, chez une autre compagnie à laquelle une création a été vendu.

Liste des tables

- Spectacle (id_spectacle, nom_spectacle, tarif_plein, tarif_réduit, rentabilité);
- Compagnie (id_compagnie, nom_compagnie, metteur_en_scène);
- Représentation (id_représentation, Pol, date_début, date_représentation, nb_places_max, n_tarif_plein, nb_tarif_réduit, nb_places_réservées, durée, gain, id_spectacle);
- Création (id_spectacle);
- Achat (id_spectacle, date_achat, valeur, id_compagnie);
- Vente (id_vente, date_vente, valeur, id_spectacle, id_compagnie);
- Subvention (id_subvention, nom_mécène, valeur, id_spectacle);
- Dépenses (id_dépense, date_dépense, description, coût, id_spectacle);
- Interne (id_représentation);
- Externe (id_représentation, lieu, ville, pays);
- Réservation (id_réservation, date_réservation, date_limite, id_représentation);
- Billetterie (entrée, date_entrée, tarif, prix, id_représentation);

À la fin de ce rapport, vous pourrez trouver les diagrammes MCD & MLD.

Fonctions

Pour assurer le bon fonctionnement de la base de données, nous avons défini plusieurs triggers et fonctions dont voici une liste avec une brève explication de leur comportement.

- fonction clean() → à lancer avant toute autre action afin de supprimer les réservations périmées.

- Fonction update_reservation() → rend la base de données cohérente après la fonction clean(). Met à jour le nombre de place réservées pour chaque représentation.

- fonction buyFullPrice(n INTEGER, i INTEGER) → permet d'acheter n billet à plein tarif pour la représentation i.

- fonction buyHalfPrice(n INTEGER, i INTEGER) → permet d'acheter n billet à plein réduit pour la représentation i.

- fonction multipleReservation(n INTEGER, i INTEGER) → permet de réserver n billet pour la représentation i.

- fonction check_création() → vérifie que le spectacle que l'on veut une création n'est pas un spectacle qui a été acheté à une compagnie.

- fonction check_achat() → vérifie que le spectacle que l'on veut une pièce achetée n'est pas une création du Théâtre du Chrysanthème.

- fonction check_interne() → vérifie qu'une représentation n'a pas déjà été déclarée en représentation externe.

- fonction check_externe() → vérifie qu'une représentation n'a pas déjà été déclarée en représentation interne.

- fonction politique(choix INTEGER, prix REAL, date_b DATE, date_f DATE, max INTEGER, n INTEGER) → permet de calculer le prix d'une place pour une représentation en fonction de la date courante et de la politique tarifaire choisie pour cette représentation.

- fonction check_place() → vérifie qu'il reste de la place.

- fonction add_place_plein() → rend la base de données cohérente lors de l'achat d'une place au tarif plein. Ajoute une entrée au tarif plein dans la table 'Billetterie', additionne le prix du billet dans le gain de la représentation et de la rentabilité du spectacle.

- fonction add_place_réduit() → rend la base de données cohérente lors de l'achat d'une place au tarif plein. Ajoute une entrée au tarif réduit dans la table 'Billetterie', additionne le prix du billet dans le gain de la représentation et de la rentabilité du spectacle.

- fonction check_place_réservation() → vérifie que la date de la demande de réservation est toujours valide (avec une date limite pour les réservations) et qu'il reste de la place.

- fonction add_réservation() → rend la base de données cohérente lors de la prise d'une réservation. Ajoute une entrée dans la table 'Réservation'.

- fonction unmake_réservation() → rend la base de données cohérente lors de l'annulation d'une réservation. Enlève une entrée dans la table 'Réservation'.

- fonction make_subvention() → rend la base de données cohérente lors d'une subvention. Ajoute le gain de la subvention à la rentabilité du spectacle.

- fonction unmake_subvention() → rend la base de données cohérente lors de l'annulation d'une subvention. Enlève le prix de la subvention à la rentabilité du spectacle.

- fonction check_appartenance() → vérifie que la pièce est l'une des créations du Théâtre du Chrysanthème. On ne peux pas faire de dépenses sur une pièce que l'on a acheté et on ne également pas la vendre.

- fonction make_dépense() → rend la base de données cohérente lors d'une dépense pour un spectacle. Enlève le prix des dépenses à la rentabilité du spectacle.

- fonction unmake_dépense() → rend la base de données cohérente lors de l'annulation d'une dépense pour un spectacle. Retourne la rentabilité du spectacle à la normale.

- fonction make_vente() → rend la base de données cohérente lors d'une vente d'un spectacle. Ajoute le prix de la vente à la rentabilité du spectacle.

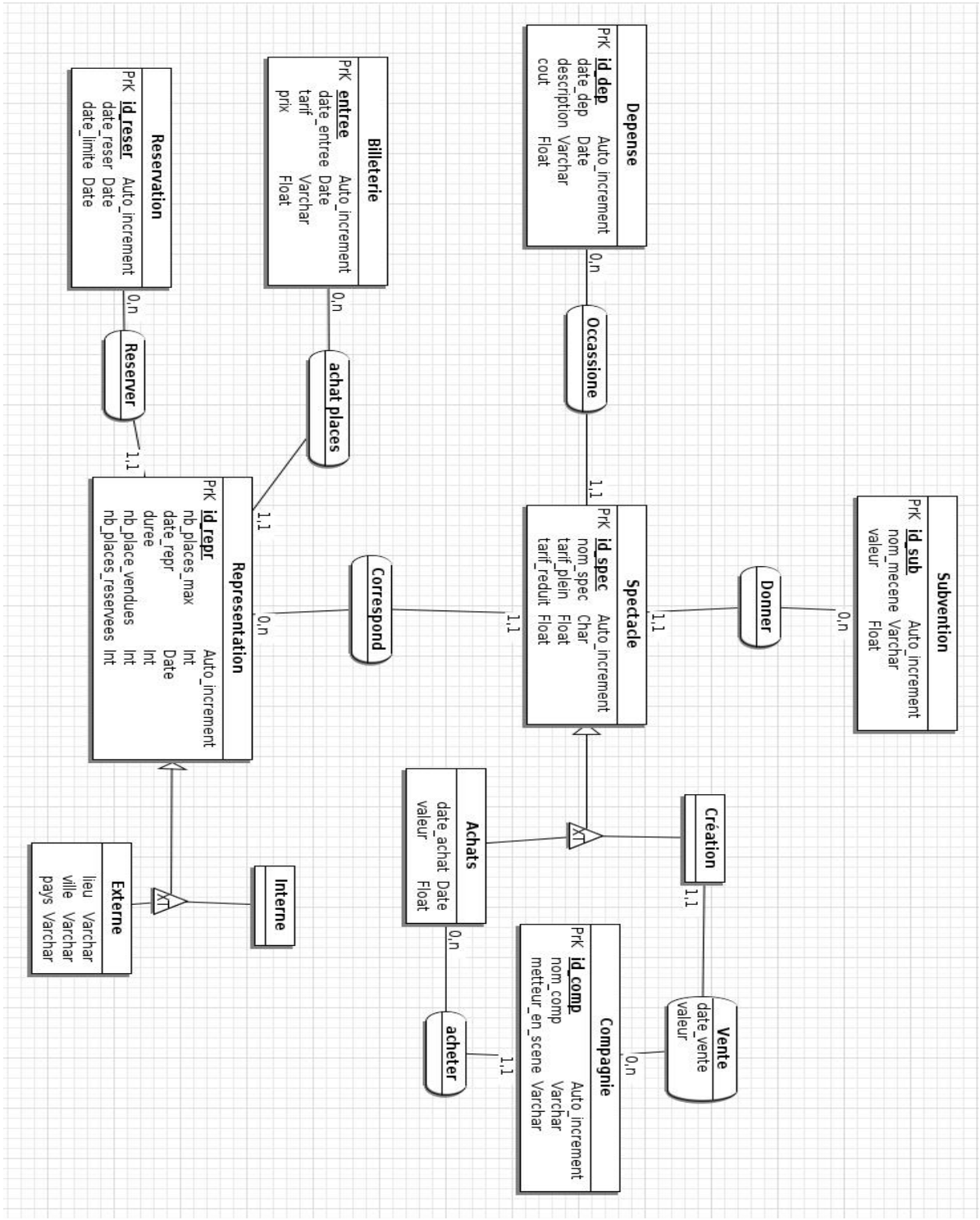
- fonction unmake_vente() → rend la base de données cohérente lors de l'annulation d'une vente d'un spectacle. Enlève le prix de vente à la rentabilité du spectacle.

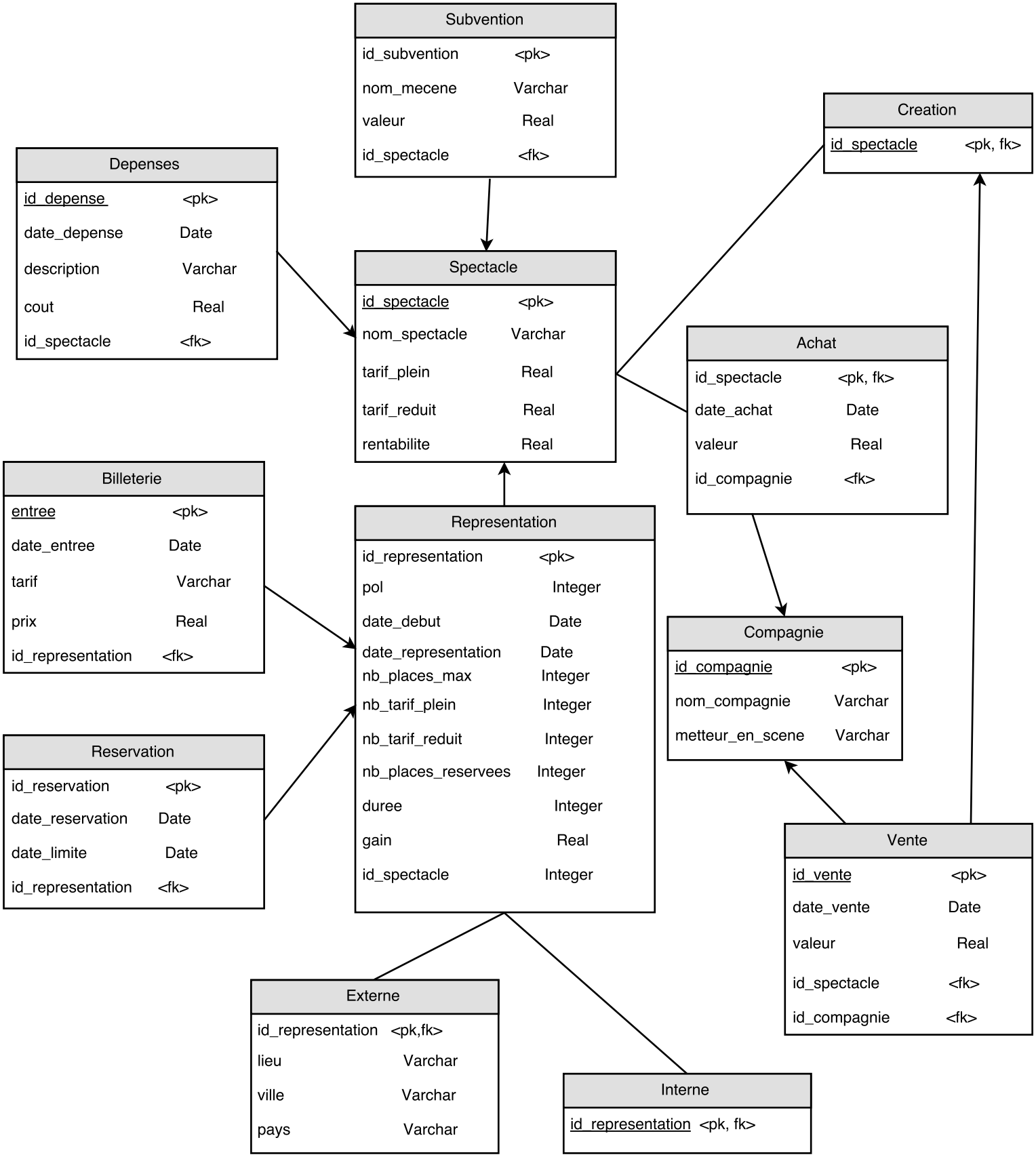
- fonction make_achat() → rend la base de données cohérente lors de l'achat d'un spectacle. Soustrait à la rentabilité du spectacle, le prix pour lequel on l'a acheté.

- fonction unmake_achat() → rend la base de données cohérente lors de l'annulation de l'achat d'un spectacle. Retourne la rentabilité du spectacle à la normale.

- fonction give_refound() → permet de faire un remboursement d'un ticket. Libère une place pour la représentation et enlève le prix du billet à la rentabilité et au gain.

Diagramme de classes & Passage aux tables




Subvention
id_subvention <pk>
nom_mecene Varchar
valeur Real
id_spectacle <fk>
Creation
id_spectacle <pk, fk>
Depenses
id_depense <pk>
date_depense Date
description Varchar
cout Real
id_spectacle <fk>
Spectacle
id_spectacle <pk>
nom_spectacle Varchar
tarif_plein Real
tarif_reduit Real
rentabilite Real
Achat
id_spectacle <pk, fk>
date_achat Date
valeur Real
id_compagnie <fk>
Billeterie
entree <pk>
date_entree Date
tarif Varchar
prix Real
id_representation <fk>
Representation
id_representation <pk>
pol Integer
date_debut Date
date_representation Date
nb_places_max Integer
nb_tarif_plein Integer
nb_tarif_reduit Integer
nb_places_reservees Integer
duree Integer
gain Real
id_spectacle Integer
Compagnie
id_compagnie <pk>
nom_compagnie Varchar
metteur_en_scene Varchar
Reservation
id_reservation <pk>
date_reservation Date
date_limite Date
id_representation <fk>
Vente
id_vente <pk>
date_vente Date
valeur Real
id_spectacle <fk>
id_compagnie <fk>
Externe
id_representation <pk,fk>
lieu Varchar
ville Varchar
pays Varchar
Interne
id_representation <pk, fk>