

Modèle Logique de Données (MLD)

Héritage

Dans cette partie nous allons transformer tous les héritages de notre MCD en utilisant les notions vues en cours :

Contributeur

Héritage par classe mère car il s'agit d'un héritage complet :

- Contributeur (#id_Contrib : integer, Nom : Varchar, Prénom : Varchar, Type : {Producteur, Réalisateur})

Contraintes :

- CHECK (Type = 'Producteur' OR Type = 'Adhérent')

Associations :

- Contribue (#IdContrib => Contributeur, #IdFilm => Film)

Vues

- *Contributions* : Vue nous montrant tous films avec leurs contributeurs et leurs rôles respectifs

Normalisation

La table est du type 2NF avec la dépendance fonctionnelle suivante :

- Id_Contrib -> Nom, Prénom, Type

Produit

Héritage par classe mère car il s'agit d'un héritage complet :

- Produits (#Id_Prod : Integer, Nom: varchar, Prix : Float, Type : {Boisson, Alimentaire})

Contraintes :

- CHECK (Type = 'Boisson' OR Type = 'Alimentaire')
- CHECK (Prix > 0)
- CHECK (Nom NOT NULL)

Normalisation

La table est du type 2NF avec la dépendance fonctionnelle suivante :

- Id_Prod -> Nom, Prix, Type

Spectateur

Héritage par classe mère car il s'agit d'un héritage presque complet :

- Spectateur (#Id_Spec : Varchar, Nom : Varchar, Prénom : Varchar, Adresse : varchar, Dernier_Rechargement : Date, Solde : Integer, Type {Normal, Abo}, #IdProj => Séance)

Contraintes :

- CHECK (Id_Spec NOT NULL AND UNIQUE)
- CHECK (Type = 'Normal' OR Type = 'Abo')
- CHECK (NOT ((Type = 'Normal') AND ((Adresse_mail NOT NULL) OR (Dernier_Rechargement NOT NULL) OR (SOLDE NOT NULL)))
- CHECK ((Type = 'Abo') AND ((Adresse_mail NOT NULL) OR (Dernier_Rechargement NOT NULL) OR (SOLDE NOT NULL))
- CHECK ((Type = 'Abo') AND ((Adresse_mail NOT NULL) OR (Dernier_Rechargement > Date_Aujourd'hui) OR (SOLDE <= 0))

Vues :

- *Adhérent* : Vue nous montrant les adhérents avec leurs nom, prénom et leur solde restant
- *Non-Adhérent* : Vue nous montrant les spectateurs normaux avec leurs nom, prénom et le tarif payé

Normalisation

La table est du type 2NF avec la dépendance fonctionnelle suivante :

- Id_Spec -> Nom, Prénom, Adresse

Entrée

Héritage par classe mère car il s'agit d'un héritage presque complet :

- Entrée (#Num_Entrée: Integer, Type : {Abonnement, Tarif_Adulte, Tarif_Enfant, Tarif_Etudiant, Tarif_Dimanche}, #Numero => Salle)

Contraintes

- CHECK (Type = 'Abonnement' OR Type = 'Tarif_Adulte' OR Type = 'Tarif_Enfant' OR Type = 'Tarif_Etudiant' OR Type = 'Tarif_Dimanche')
- CHECK (Num_Entrée NOT NULL)

Vues

- *Entrées* : Vue nous montrant le nombre de places achetées pour chaque salle

Normalisation

La table est du type 2NF avec la dépendance fonctionnelle suivante :

- Num_Entrée -> Type, Numero

Autres Relations

Distributeur

Il s'agit ici de transformer la composition "Distribue", on ajoute donc une clé étrangère dans "Distributeur" qui nous renverra vers "Films" :

- Distributeur (#Id_Distrib : Integer, Nom : Varchar)

Contraintes

- CHECK (Nom UNIQUE AND NOT NULL)

Vue

- *Distribution* : Vue nous montrant une liste des films et leurs distributeurs respectifs

Normalisation

La table est du type 2NF avec la dépendance fonctionnelle suivante :

- Id_Distrib -> Nom

Séance

Il s'agit ici de transformer l'association "Projetter", on ajoute donc une clé étrangère dans "Séance" qui nous renverra vers "Film"

- Séance (#Id_Proj : Integer, Date : Date, Heure : heure, Langue : { VO, VOSTFR, VF}, #IdFilm : Varchar => Film, #IdSalle => Salle)

Contraintes

- CHECK (Id_Proj NOT NULL AND UNIQUE)
- CHECK ((Langue = 'VO') OR (Langue = 'VOSTFR') OR (Langue = 'VF'))

Vues

Séance_Prévues : Vue nous montrant les séances prévues avec le numéro de la salle, le film projeté et l'heure de projection

Normalisation

La table est du type 2NF avec la dépendance fonctionnelle suivante :

- Id_Proj -> Date, Heure, Langue, Film, Salle

Salle

Il s'agit ici de transformer l'association "Projeté dans", on ajoute donc une clé étrangère dans "Salle" qui nous renverra vers "Séance" :

- Salle (#Numéro : Integer, Nbr_Places : Integer)

Contraintes

- CHECK (Numéro UNIQUE AND NOT NULL)
- CHECK (Nbr_Places >= 0)

Normalisation

La table est du type 2NF avec la dépendance fonctionnelle suivante :

- Numéro -> Nbr_Places

Note_Spec

Il s'agit ici de transformer deux associations, l'association "Moyenne" et l'association "Note_Donne", soit une association 1--N et une autre du type 1--1 :

- Note_Spec (Note : Integer, #IdFilm : Varchar => Film, IdSpec => Spectateur)

Contraintes

- CHECK (Id_Spec UNIQUE)
- CHECK (Note = 0 OR 1 <= Note <= 5)

Vues

- *Note_Donné* : Vue nous montrant le nom, prénom du spectateur, la note qu'il à donné et le film qu'il a noté

Normalisation

La table n'a pas de normalisation car pas de DF

Film

- Film (#Id_Film : Integer, Titre : Varchar, Date_Sortie : Date, Age_Min : Integer, Genre : { Horreur, Policier, Thriller, Comique, Animation }, #IdDistrib => Distributeur)

Contraintes

- CHECK ((Genre = 'Horreur') OR (Genre = 'Thriller') OR (Genre = 'Policier') OR (Genre = 'Comique') OR (Genre = 'Animation')))
- CHECK (Id_FILM UNIQUE)

Normalisation

La table est du type 1NF avec la dépendance fonctionnelle suivante :

- Id_Film -> Titre, Date_Sortie, Age_Min, Genre

Vendeur

- Vendeur (#Id_Cine : Integer, Nom : Varchar, Prénom : Varchar)

Association

Comme il s'agit d'une association 1..* -- 0..N entre Vendeur et Produits, Vendeur et Entrée, Spectateur et Produits et enfin Spectateur et Entrée. On peut créer une nouvelle table nommée Vente, qui nous aidera à traquer toutes les ventes, avec les clés primaires et étrangères suivante :

- Vente (#Id_Vente : Integer, #Id_Spec => Spectateur, #IdCiné => Vendeur, #IdProd => Produit, #NumEntree => Entrée)

Contraintes

- CHECK (Id_Cine UNIQUE)
- CHECK Prénom NOT NULL
- CHECK Nom NOT NULL

Vues

- Vente : Vue nous montrant les différentes ventes avec le nom du vendeur, le nom du spectateur, le nom du produit et le prix du produit

Normalisation

La table est du type 2NF avec la dépendance fonctionnelle suivante :

- Id_Cine -> Nom, Prenom

Rôles

Création des rôles :

- Informaticien
- Spectateur_Adhérent;
- Manager;
- Vendeur;

Attribution de droits :

Information

- ALL sur Vendeur
- ALL sur Film, Contributeur, Distributeur, Salle, Séance, Entrée, Contribue, Spectateur, Produit, Note_Spec, Vente

Spectateur Adhérent

- SELECT sur Film, Séances_Prévues, Distribution, Spectateur TO Spectateur_Adhéren

Vendeur

- SELECT sur Produit, Séances_Prévues, Vendeur, Vente
- UPDATE sur Spectateur, Adhérents, Non_Adhérent, Produits, Vente

Manager

- UPDATE sur Vendeur, Séance, Film
- SELECT sur Vente, Produits, Adhérents, Non_Adhérent