

OUEDRAOGO Taoufiq
VERGNOL Calliste
HAFIZ Adam

MLD

Client(#tel : Integer, nom : String, adresse : String)

Compte(#date_crea : DateTime, balance : Float, statut : {ouvert, bloqué, fermé})

Asso_Compte_Client((#tel => Client.tel, #date_crea => Compte.date_crea)

Contraintes :

PROJECTION(Client, tel) = PROJECTION(Asso_Compte_Client, tel) AND
PROJECTION(Compte, date_crea) = PROJECTION(Asso_Compte_Client, date_crea)

CompteEpargne(#date_crea => Compte.date_crea, balance => Compte.balance, solde_min_const : float) avec
{balance > solde_min_const}

CompteRevolving(#date_crea => Compte.date_crea, balance => Compte.balance, taux_j : float, montant_min : float) avec {0 < taux_j < 1, montant_min < 0, et balance < montant_min}

CompteCourant(#date_crea => Compte.date_crea, balance => Compte.balance, montant_decouvert_autorise : Float, date_debut_decouvert : Date, max_solde : Float, min_solde : Float) avec
{max_solde > 0, min_solde > 0, min_solde < balance < max_solde,

date_debut_decouvert NULLABLE, montant_decouvert_autorise
NULLABLE}

Contraintes :

PROJECTION(Compte, date_crea) = PROJECTION(CompteCourant,
date_crea) UNION PROJECTION(CompteRevolving, date_crea) UNION
PROJECTION(CompteEpargne, date_crea)

Contraintes sur l'exclusivité de l'héritage :

INTERSECTION (PROJECTION(CompteCourant, date_crea),
PROJECTION(CompteEpargne, date_crea),
PROJECTION(CompteRevolving, date_crea)) = {}

Operation(#id : Integer, montant : Float, date : Date, etat : {traité,
non_traité}, client => Client.tel, date_crea => Compte.date_crea) avec
{(client, date_crea) UNIQUE et montant > 0}

CarteBleu(#id => Operation.id, compteRevolving =>
CompteRevolving.date_crea, compteCourant =>
CompteCourant.date_crea) avec {(compteCourant, compteRevolving)}
UNIQUE et (compteRevolving OR compteCourant)

EmissionCheque(#id => Operation.id, compteRevolving =>
CompteRevolving.date_crea, compteCourant =>
CompteCourant.date_crea) avec {(compteCourant, compteRevolving)}
UNIQUE et (compteRevolving OR compteCourant)

DepotCheque(#id => Operation.id, compteRevolving =>
CompteRevolving.date_crea, compteCourant =>
CompteCourant.date_crea) avec {(compteCourant, compteRevolving)}
UNIQUE et (compteCourant OR compteRevolving)

Virement(#id => Operation.id, compteCourant =>
CompteCourant.date_crea, compteRevolving =>
compteRevolving.date_crea, compteEpargne =>
compteEpargne.date_crea) avec {(compteCourant, compteRevolving,

compteEpargne)} UNIQUE et (compteCourant OR compteRevolving OR compteEpargne)

CreditGuichet(#id => Operation.id, compteCourant => CompteCourant.date_crea, compteRevolving => compteRevovling.date_crea, compteEpargne => compteEpargne.date_crea) avec {(compteCourant, compteRevolving, compteEpargne)} UNIQUE et (compteCourant OR compteRevolving OR compteEpargne)

DebitGuichet(#id => Operation.id, compteCourant => CompteCourant.date_crea, compteRevolving => compteRevovling.date_crea, compteEpargne => compteEpargne.date_crea) avec {(compteCourant, compteRevolving, compteEpargne)} UNIQUE et (compteCourant OR compteRevolving OR compteEpargne)

Contraintes :

PROJECTION(Operation, id) = PROJECTION(CarteBleu, id) UNION
PROJECTION(EmissionCheque, id) UNION PROJECTION(DepotCheque, id) UNION PROJECTION(Virement, id) UNION
PROJECTION(CreditGuichet, id) UNION PROJECTION(DebitGuichet, id)

Contraintes sur l'exclusivité de l'héritage :

INTERSECTION (PROJECTION(CarteBleu, id),
PROJECTION(EmissionCheque, id), PROJECTION(DepotCheque, id),
PROJECTION(Virement, id), PROJECTION(CreditGuichet, id),
PROJECTION(DebitGuichet, id)
) = {}