

Modèle Logique De Données

Justification de transformation de l'héritage :

- Concernant la classe Client, la classe mère est abstraite, l'héritage est non exclusif et l'héritage n'est pas complet, la seule possibilité est de transformer notre héritage par référence.

Client(#Pseudo : string, Nom : string, Prénom : string, Âge : date, Photo : blob, Coordonnées : string)

Propriétaire(#pseudo => Client)

Locataire(#pseudo => Client, NumPermis : string, ExpirePermis : date) avec NumPermis key

Entreprise(#pseudo => Client, NumSiren : int, ListeConducteur : Conducteur), avec NumSiren key

Contrainte : PROJECTION(Client, Pseudo) =
PROJECTION(Locataire, pseudo) UNION PROJECTION(Entreprise, pseudo)
UNION PROJECTION(Propriétaire, pseudo)

Véhicule(#N°Immat : string, Catégorie : string, Marque : string, Modèle : string, Couleur : string, Options : string, Carburant : enum{'Essence', 'Diesel', 'Gaz', 'Électrique'}, AnnéeMiseCirculation : date, Kilométrage : integer, Description : string, contratA => ContratAssurance, responsable => ResponsableQualité, propriétaire => Propriétaire, avis => Avis) avec contratA NOT NULL, responsable NOT NULL, propriétaire NOT NULL, Kilométrage > 0

Contrainte : PROJECTION(Véhicule, contratA) =
PROJECTION(ContratAssurance, NumContratA)

LocationPL(#propriétaire => Propriétaire, #locataire => Locataire)

LocationLV(#véhicule => Véhicule, #locataire => Locataire)

Contraintes : PROJECTION(LocationLV, véhicule) =
PROJECTION(Véhicule, N°Immat) AND PROJECTION(LocationLV, locataire)
= PROJECTION(Locataire, pseudo)

LocationEV(#véhicule => Véhicule, #entreprise => Entreprise)

Signaler(#signalement => Signalement, #locataire => Locataire)

PropriétéLoc(#propriétaire => Propriétaire, #véhicule => Véhicule,
DateDebutLoc : date, DateFinLoc : date, SeuilKm : int, SeuilCarbu :
int, ListePays : Pays) avec SeuilKm > 0, SeuilCarbu > 0,
DateDebutLoc NOT NULL, DateFinLoc NOT NULL

Contraintes : PROJECTION(PropriétéLoc, propriétaire) =
PROJECTION(Propriétaire, pseudo) AND PROJECTION(PropriétéLoc,
véhicule) = PROJECTION(Véhicule, N°Immat)

Avis(#véhicule => Véhicule, contrat => ContratLocation, Note : int,
Commentaire : text) avec contrat KEY, 0 <= Note <= 5

ContratLocation(#NumContratL : int, DateDébutCL : date, véhicule =>
Véhicule) avec véhicule NOT NULL, DateDébutCL NOT NULL

Check-out(#contrat => ContratLocation, NbDégâtsOut : int,
kilometrage: int, niveauCarburant : int, EtatProp :
enum{'Excellent', 'Bon', 'Passable', 'Moyen', 'Médiocre', 'Mauvais',
'Exécrable'}) avec NbDégâtsOut >=0

Check-in(#contrat => ContratLocation, NbDégâtsIn : int, kilométrage
: int, NiveauCarburant : int, EtatLoc : enum{'Excellent', 'Bon',
'Passable', 'Moyen', 'Médiocre', 'Mauvais', 'Exécrable'}, PhotoLoc :
blob) avec NbDégâtsIn >= 0

Facture(#contratLocation => ContratLocation, Montant : int,
Kilométrage : int, DateFacture : date, MoyenDePaiement : enum{'Carte
Bancaire', 'Chèque', 'Espèce'}) avec Montant > 0, Kilométrage > 0,
DateFacture NOT NULL

Contraintes: Projection(Facture, contratLocation) =
Projection(ContratLocation, NumContratL)

Signalement(#véhicule => Véhicule, NbSignalement : int, responsable
=> ResponsableQualité) avec responsable NOT NULL, NbSignalement =
NbSignalement%4

ÉmetLS(#signalement => Signalement, #locataire => Locataire)

ResponsableQualité(#id : int, AnalyseÉtat : enum{'Excellent', 'Bon',
'Passable', 'Moyen', 'Médiocre', 'Mauvais', 'Exécrable'})

ContratAssurance(#NumContratA : int)

Franchise(#contratAssurance => ContratAssurance, ChoixFranchise :
enum {'Sans réduction', 'Franchise réduite', 'Zéro Franchise'})

Contrainte :

Projection(Franchise, contratAssurance) =
Projection(ContratAssurance, NumContratA)