

# MLD Tables Normalisées

On choisit les tables "Véhicule", "Locataire", "Check-in" et "Check-out" à normaliser. En effet, ce sont les tables pour lesquelles nous avons décelé des dépendances fonctionnelles entre les attributs autre que celle fournie par la clé de la relation.

- **Véhicule** → Dépendances Fonctionnelles :
  - N°Immat → Catégorie, Marque, Modèle, Couleur, Options, Carburant, AnnéeMiseCirculation, Kilométrage, Description, contratA, responsable, propriétaire, avis
  - Marque → Modèle
  - Modèle → Marque
- **Check-in** → Dépendances Fonctionnelles :
  - contrat → NbDégâtsIn, kilométrage, NiveauCarburant, EtatLoc, PhotoLoc
  - NbdégâtsIn → EtatLoc
- **Check-out** → Dépendances Fonctionnelles :
  - contrat → NbDégâtsOut, kilometrage, niveauCarburant, EtatProp
  - NbdégâtsOut → EtatProp
- **Locataire** → Dépendances Fonctionnelles :
  - Pseudo → Nom, Prénom, Age, Photo, Coordonnées (héritées de la table Client)
  - Numpermis → ÉmissionPermis, ExpirePermis

Ainsi on obtient le modèle logique de données mis à jour pour inclure la normalisation de nos tables. Celles-ci sont écrites en vert.

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

**Propriétaire**(#pseudo => Client)

**Locataire**(#pseudo => Client, permis => Permis)

**Permis**(#Numpermis : string, EmissionPermis : Date, ExpirePermis : DATE) avec EmissionPermis < ExpirePermis

**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)**

Dans le cas de la relation véhicule, nous sommes confronté à un problème. En transformant cette relation en 3NF, à cause de la double dépendance entre Marque et Modèle, les deux attributs sont clé candidates. Néanmoins avec cette représentation, une marque ne peut pas avoir plusieurs modèles. Nous sommes contraints d'ajouter un id pour avoir la possibilité de définir plusieurs modèles pour une marque.

**Véhicule**(#N°Immat : string, Catégorie : 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, marquemodèleid => MarqueModèle) avec contratA NOT NULL, responsable NOT NULL, propriétaire NOT NULL, marquemodèle NOT NULL, Kilométrage > 0

**MarqueModèle**(#Marque : string, #Modèle : string)

**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)

**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-in**(#contrat => ContratLocation, kilométrage : int,  
NiveauCarburant : int, PhotoLoc : blob, dégâtsinetat =>  
DégâtsInEtat) avec NbDégâtsIn >= 0

**DégâtsInEtat**(#NbDégâtsIn : integer, EtatLoc : Etat)

**Check-out**(#contrat => ContratLocation, kilometrage: int,  
niveauCarburant : int,dégâtsoutetat => DégâtsOutEtat) avec  
NbDégâtsOut >=0

**DégâtsOutEtat**(#NbDégâtsOut : integer, EtatProp : Etat)

**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)**