# 

# Dossier MarqueOpen

Table des matières

[Dossier MarqueOpen 1](#_Toc469431486)

[Questions 3](#_Toc469431487)

[Description des tables 3](#_Toc469431488)

[Table Véhicule 3](#_Toc469431489)

[Table Stockage 4](#_Toc469431490)

[Table TradLibPiece 5](#_Toc469431491)

[Table Provenance 6](#_Toc469431492)

[Schéma EA corrigé et final 6](#_Toc469431493)

[Transformations Avant – Après 8](#_Toc469431494)

[9](#_Toc469431495)

[Première transformation 10](#_Toc469431496)

[Deuxième transformation 11](#_Toc469431497)

[Troisième transformation 12](#_Toc469431498)

[Quatrième transformation 12](#_Toc469431499)

[Contraintes d’intégrité 14](#_Toc469431500)

[Script de création **Erreur ! Signet non défini.**](#_Toc469431501)

[Contraintes additionnelles 14](#_Toc469431502)

[Schéma relationnel 14](#_Toc469431503)

## Questions

1. Doit-on faire de modèle une entité ?
   1. Oui car la motorisation et le cylindré dépendent du Modèle, de plus on a une option de vérification des modèles
2. Quel serait l’id naturel du véhicule ?
   1. Pas la plaque minéralogique dans tous les cas car la voiture peut changer de propriétaire dans le temps et donc la plaque aussi.
3. Quel serait l’id naturel de visite ?
   1. N°visite et la plaque d’entrée
4. Doit-on lier facture à un véhicule ou à une visite ?
   1. On la lie à visite car une visite ne concerne qu’un seul véhicule.
5. Doit-on faire un héritage pour véhicule dont les sous types sont Voiture et camion ?
   1. Oui mais on ne le représente pas en conceptuelle, on a utilisé le type générique
6. Doit-on mettre 1-1 ou 1-N du coté TypePièce ?
   1. Désaccord , 1-1 car un type de pièce ne peut être stocké que dans un lieu de stockage, en effet, toutes les pièces sont centralisées et stockées au même endroit.
   2. 1-N : Un type de pièce peut être stocké dans plusieurs endroits de stockage si jamais la place disponible pour le stock n’est pas suffisante, obligation de mettre le stock ailleurs.

## Description des tables

### Table Véhicule

|  |  |  |
| --- | --- | --- |
| Type d’entité | **VEHICULE**  Représente un véhicule répertorié au garage  Tant que le propriétaire du garage juge censé de l’enlever | |
| Définition |
| Durée de vie |
| ***Attributs*** | ***Définition*** | ***Domaine*** |
| IdVéhicule | Id artificiel du véhicule permettant d’identifier le véhicule de manière unique | Integer auto incrementable |
| NumeroMineralogique | Plaque minéralogique du véhicule | Alphanumérique(9 car) |
| NumeroChassis | Numéro de châssis assigné au véhicule. Identifiant naturel, unique pour chaque véhicule présent dans la BD | Alphanumérique(35 car) |
| Marque | Marque de la voiture | Alphanumérique(20 car) |
| Description | Brève description contenant diverses informations utiles | Alphanumérique(50 car) |
| Couleur | Couleur du véhicule | Alphanumérique(25 car) |
| ChargeMax | Charge utile maximum pour un camion | Integer de 0 à 100 |
| DateContrôle | Date du dernier contrôle technique pour un camion | Date valide  Date valide > AnnéeFabrication |
| TypeJante | Type de jante du véhicule en question | Alphanumérique(25 car) |
| TypeVéhicule | Type de véhicule, soit Camion ou Voiture | Alphanumérique(20 car) |

### Table Stockage

|  |  |  |
| --- | --- | --- |
| Type d’entité | **Stockage**  Lieu de stockage où peuvent être entreposés les différents types de pièces  Jusqu’à ce que le garagiste décide de vendre ou rendre son espace de stockage | |
| Définition |
| Durée de vie |
| ***Attributs*** | ***Définition*** | ***Domaine*** |
| CodeStockage | Code de stockage unique identifiant un lieu, une étagère, une pièce | Integer |
| QuantitéStock | Quantité maximale possible de stock | Integer > 0 |
| DateDernierStock | Date du dernier relevé de stock | Date |
| QuantitéStockDate | Quantité du stock à une date souhaitée | Date |

### Table ouvrier

|  |  |  |
| --- | --- | --- |
| Type d’entité | **Ouvrier**  La personne qui va s’occuper des éventuelles réparations.  Une fois que le CEO décide de se séparer de l’ouvrier en question. | |
| Définition |
| Durée de vie |
| ***Attributs*** | ***Définition*** | ***Domaine*** |
| matricule | Code de l’ouvrier unique identifiant un ouvrier | Integer auto increment |
| nom | Nom de l’ouvrier | Alphanumérique(20 car) |
| Prenom | Prenom de l’ouvrier | Alphanumérique(20 car) |
| tel | Numero de telephone de l’ouvrier(minimum un) | Alphanumérique(10 car) |
| qualification | Degré du niveau de compétence de l’ouvrier : 3 possiblité(Maitrsie,Debutant ou Confirmé) | Alphanumérique(1 car) |

### TA TradLibPiece

|  |  |  |
| --- | --- | --- |
| Type d’association | TradLibPiece  Une pièce est traduite dans une langue donnée  Tant que l’on décide de traduire les pièces dans le garage | |
| Définition |
| Durée de vie |
| Défini sur | Pièce | Langue |
| Rôles | Possède | Traduit |
| Cardinalités | 0-N | 0-N |
| Attributs | Définition | Domaine |
| IdTrad | Identifiant de la traduction de la pièce | Integer |
| LibellePiece | Traduction de la pièce dans une langue donnée | Varchar |

### TA Provenance

|  |  |  |
| --- | --- | --- |
| Type d’association | Provenance  Un véhicule est vendu par un concessionnaire  Tant que le véhicule existe dans la BD | |
| Définition |
| Durée de vie |
| Défini sur | Concessionnaire | Véhicule |
| Rôles | Vend | AétéVendu |
| Cardinalités | 0-N | 1-1 |
| Attributs | Définition | Domaine |
|  |  |  |

## Schéma EA corrigé et final



## Transformations Avant – Après

## 

### Première transformation

|  |  |
| --- | --- |
| Avant | Après |
|  |  |

Nous avons ajouté un type d’opération pour distinguer les différentes opérations que chaque ouvrier pourrait effectué (le type d’opération est relié à plusieurs Ouvrier et un ouvrier peut réaliser plusieurs types d’opérations, non représenté sur le schéma EA après). Le type opération permet de renseigner également des données théoriques telles que la durée de l’opération.

Nous avons également enlevé l’entité type de pièce pour ne garder que l’entité pièce, les pièces ne sont pas triées pas catégories, il n’y a donc aucun intérêt à représenter l’entité type de pièce.

### Deuxième transformation

|  |  |
| --- | --- |
| Avant | Après |
|  |  |

Nous avons voulu par la suite représenter les valeurs distinctes en guise de vérification de la qualification de l’ouvrier et des numéros de téléphone

### Troisième transformation

|  |  |
| --- | --- |
| Avant | Après |
|  |  |

Pouvoir représenter les instances des téléphones.

### Quatrième transformation

|  |  |
| --- | --- |
| Avant | Après |
|  |  |

Nous devons transformer la TA « appartenance » en TE pour vérifier que la visite concerne bien une propriété donnée(CA). Dans le cas où le TE Vis\_app n’existera pas, un propriétaire pourrait passer une visite avec un véhicule qui lui appartient mais pourrait faire payer un ancien possesseur de ce véhicule étant donné que la visite ne vérifie pas que c’est la bonne propriété (appartenance) concernant cette visite .

## Contraintes d’intégrité



Nous devrons avertir le SGBD que le 1-N est en réalité une 1-2.

## Contraintes additionnelles

CA1 : le type du véhicule ne peut être que « camion » ou « voiture »

CA2 : La propriété ne peut pas être périmée par rapport à la date du devis et de la visite

CA3 : Vérifier que la motorisation et la cylindrée dépendent du modèle

CA4 : Pas de chevauchement de date entre les propriétés (impossible d’avoir plusieurs propriétaire pour un même véhicule au même moment)

## Schéma relationnel

|  |  |  |  |
| --- | --- | --- | --- |
|  | Résumé contrainte | Prise en charge SGBD | Technique |
| CA1 | Type du véhicule | non | Check |
| CA2 | La propriété ne doit pas être périmée à la date du devis ou de la visite | Oui | Check |
| CA3 | Vérification de la motorisation et cylindrée | Oui | Trigger ou bien valeurs distinctes |
| CA4 | Chevauchement | Oui | Check |
| CAR1 | 1-N du coté propriétaire | Limite acceptée | Ne sera pas contrôlée |



## Script de création de table

-- \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

-- \* Standard SQL generation

-- \*--------------------------------------------

-- \* DB-MAIN version: 9.3.0

-- \* Generator date: Feb 16 2016

-- \* Generation date: Tue Dec 13 22:06:19 2016

-- \* LUN file: C:\Users\dark-\Desktop\CoursGithub\3IG\BD\MarqueOpenRel.lun

-- \* Schema: MarqueOpen/SQL

-- \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

-- Database Section

-- \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

create database MarqueOpen;

-- DBSpace Section

-- \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

-- Tables Section

-- \_\_\_\_\_\_\_\_\_\_\_\_\_

create table appartenance (

IdPropriete numeric(5) not null,

dateFin date,

dateDeb date not null,

plaqueMineralogique varchar(9) not null,

DateImmatriculation date not null,

idVehicule numeric(9) not null,

id numeric(5) not null,

constraint ID\_appartenance\_ID primary key (IdPropriete));

create table Concessionnaire (

idConcessionnaire numeric(9) not null,

nom varchar(50) not null,

Adresse varchar(80) not null,

constraint ID\_Concessionnaire\_ID primary key (idConcessionnaire));

create table Consommation (

IdConso numeric(5) not null,

Qte numeric(6) not null,

PrixReel numeric(6,2) not null,

Id numeric(5) not null,

IdOperation numeric(5) not null,

constraint ID\_Consommation\_ID primary key (IdConso));

create table Devis (

date date not null,

Numero numeric(6) not null,

IdVisite numeric(6) not null,

descriptif varchar(80) not null,

IdPropriete numeric(5) not null,

constraint ID\_Devis\_ID primary key (Numero),

constraint SID\_Devis\_Visit\_ID unique (IdVisite));

create table FactureImpaye (

idFacture numeric(5) not null,

IdVisite numeric(6) not null,

date date not null,

dateEcheance date not null,

montantAccompte numeric(8,2) not null,

solde numeric(8,2) not null,

descriptif varchar(80) not null,

montant numeric(8,2) not null,

constraint ID\_FactureImpaye\_ID primary key (idFacture),

constraint SID\_Factu\_Visit\_ID unique (IdVisite));

create table Langue (

CodeLangue numeric(5) not null,

LibelleLangue varchar(50) not null,

id numeric(5) not null,

constraint ID\_Langue\_ID primary key (CodeLangue),

constraint SID\_Langue\_ID unique (CodeLangue));

create table Ligne (

idLigne numeric(5) not null,

numeroOrdre numeric(5) not null,

Numero numeric(6) not null,

constraint ID\_Ligne\_ID primary key (idLigne));

create table LigneCommentaire (

idLigne numeric(5) not null,

texte varchar(100) not null,

constraint ID\_Ligne\_Ligne\_2\_ID primary key (idLigne));

create table LigneConsommation (

idLigne numeric(5) not null,

quantite numeric(5) not null,

PrixPieceEventuelle numeric(6,2) not null,

Id numeric(5) not null,

constraint ID\_Ligne\_Ligne\_1\_ID primary key (idLigne));

create table LigneOperation (

idLigne numeric(5) not null,

libelle varchar(80) not null,

temps numeric(5) not null,

coutHoraire numeric(3,2) not null,

coutStandard numeric(3,2) not null,

IdOperation numeric(5) not null,

constraint ID\_Ligne\_Ligne\_ID primary key (idLigne));

create table Marque (

CodeMarque numeric(5) not null,

LibelleMarque varchar(80) not null,

constraint ID\_Marque\_ID primary key (CodeMarque));

create table Modele (

idModel numeric(9) not null,

motorisation varchar(50) not null,

cylindree varchar(50) not null,

CodeMarque numeric(5) not null,

constraint ID\_Modele\_ID primary key (idModel));

create table Option (

idOption numeric(5) not null,

libelle varchar(80) not null,

descriptif varchar(80) not null,

constraint ID\_Option\_ID primary key (idOption));

create table optionNonSatandards (

idOption numeric(5) not null,

idVehicule numeric(9) not null,

constraint ID\_optionNonSatandards\_ID primary key (idVehicule, idOption));

create table optionStandards (

idModel numeric(9) not null,

idOption numeric(5) not null,

constraint ID\_optionStandards\_ID primary key (idModel, idOption));

create table Operation (

IdOperation numeric(5) not null,

n°ordre numeric(5) not null,

dureeReelle varchar(15) not null,

Code numeric(5) not null,

IdVisite numeric(6) not null,

constraint ID\_Operation\_ID primary key (IdOperation));

create table Ouvrier (

matricule numeric(5) not null,

nom varchar(50) not null,

prenom varchar(50) not null,

dateEntree date not null,

CodeQualif numeric(5) not null,

constraint ID\_Ouvrier\_ID primary key (matricule));

create table Piece (

Id numeric(5) not null,

prixCatalogue numeric(6,2) not null,

constraint ID\_Piece\_ID primary key (Id));

create table Pro\_tel (

id numeric(5) not null,

tel varchar(12) not null,

constraint ID\_Pro\_tel\_ID primary key (tel, id));

create table Proprietaire (

id numeric(5) not null,

nom varchar(80) not null,

adr\_ville varchar(50) not null,

adr\_cp numeric(5) not null,

adr\_rue varchar(30) not null,

adr\_numRue varchar(5) not null,

mail varchar(35) not null,

langue varchar(25) not null,

constraint ID\_Proprietaire\_ID primary key (id));

create table Qualification (

CodeQualif numeric(5) not null,

LibelleQualif varchar(80) not null,

constraint ID\_Qualification\_ID primary key (CodeQualif));

create table Stock (

Id numeric(5) not null,

codeStockage numeric(5) not null,

QuantiteStock numeric(5) not null,

DateDernierStock date not null,

QteStockTheorique numeric(5) not null,

constraint ID\_Stock\_ID primary key (Id, codeStockage));

create table Stockage (

codeStockage numeric(5) not null,

constraint ID\_Stockage\_ID primary key (codeStockage));

create table tel (

matricule numeric(5) not null,

tel varchar(12) not null,

constraint ID\_tel\_ID primary key (matricule, tel));

create table tel\_1 (

tel varchar(12) not null,

constraint ID\_tel\_1\_ID primary key (tel));

create table TradLibPiece (

IdTrad numeric(5) not null,

LibellePiece varchar(80) not null,

Id numeric(5) not null,

CodeLangue numeric(5) not null,

constraint ID\_TradLibPiece\_ID primary key (IdTrad));

create table TypeOperation (

Code numeric(5) not null,

descriptif varchar(80) not null,

prixUnitaire numeric(6,2) not null,

dureeTheorique varchar(15) not null,

CodeLangue numeric(5) not null,

constraint ID\_TypeOperation\_ID primary key (Code));

create table TypeOpOuv (

matricule numeric(5) not null,

Code numeric(5) not null,

constraint ID\_TypeOpOuv\_ID primary key (matricule, Code));

create table Vehicule (

idVehicule numeric(9) not null,

NumeroChassis varchar(20) not null,

description varchar(80) not null,

couleur varchar(15) not null,

chargeMax numeric(3,2),

dateControle date,

anneeFabrication date not null,

typeJante varchar(30) not null,

typeVehicule varchar(30) not null,

idModel numeric(9) not null,

idConcessionnaire numeric(9) not null,

constraint ID\_Vehicule\_ID primary key (idVehicule),

constraint SID\_Vehicule\_ID unique (NumeroChassis));

create table Visite (

IdVisite numeric(6) not null,

n°Visite numeric(6) not null,

dateEntree date not null,

dateSortie date not null,

coutTotal numeric(8,2) not null,

descriptif varchar(80),

IdPropriete numeric(5) not null,

constraint ID\_Visite\_ID primary key (IdVisite),

constraint SID\_Visite\_ID unique (n°Visite, dateEntree));

-- Constraints Section

-- \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

alter table appartenance add constraint REF\_appar\_Vehic\_FK

foreign key (idVehicule)

references Vehicule;

alter table appartenance add constraint REF\_appar\_Propr\_FK

foreign key (id)

references Proprietaire;

alter table Consommation add constraint REF\_Conso\_Piece\_FK

foreign key (Id)

references Piece;

alter table Consommation add constraint REF\_Conso\_Opera\_FK

foreign key (IdOperation)

references Operation;

alter table Devis add constraint ID\_Devis\_CHK

check(exists(select \* from Ligne

where Ligne.Numero = Numero));

alter table Devis add constraint SID\_Devis\_Visit\_FK

foreign key (IdVisite)

references Visite;

alter table Devis add constraint REF\_Devis\_appar\_FK

foreign key (IdPropriete)

references appartenance;

alter table FactureImpaye add constraint SID\_Factu\_Visit\_FK

foreign key (IdVisite)

references Visite;

alter table Langue add constraint REF\_Langu\_Propr\_FK

foreign key (id)

references Proprietaire;

alter table Ligne add constraint EQU\_Ligne\_Devis\_FK

foreign key (Numero)

references Devis;

alter table LigneCommentaire add constraint ID\_Ligne\_Ligne\_2\_FK

foreign key (idLigne)

references Ligne;

alter table LigneConsommation add constraint REF\_Ligne\_Piece\_FK

foreign key (Id)

references Piece;

alter table LigneConsommation add constraint ID\_Ligne\_Ligne\_1\_FK

foreign key (idLigne)

references Ligne;

alter table LigneOperation add constraint REF\_Ligne\_Opera\_FK

foreign key (IdOperation)

references Operation;

alter table LigneOperation add constraint ID\_Ligne\_Ligne\_FK

foreign key (idLigne)

references Ligne;

alter table Modele add constraint REF\_Model\_Marqu\_FK

foreign key (CodeMarque)

references Marque;

alter table optionNonSatandards add constraint REF\_optio\_Vehic

foreign key (idVehicule)

references Vehicule;

alter table optionNonSatandards add constraint REF\_optio\_Optio\_1\_FK

foreign key (idOption)

references Option;

alter table optionStandards add constraint REF\_optio\_Optio\_FK

foreign key (idOption)

references Option;

alter table optionStandards add constraint REF\_optio\_Model

foreign key (idModel)

references Modele;

alter table Operation add constraint REF\_Opera\_TypeO\_FK

foreign key (Code)

references TypeOperation;

alter table Operation add constraint REF\_Opera\_Visit\_FK

foreign key (IdVisite)

references Visite;

alter table Ouvrier add constraint ID\_Ouvrier\_CHK

check(exists(select \* from tel

where tel.matricule = matricule));

alter table Ouvrier add constraint REF\_Ouvri\_Quali\_FK

foreign key (CodeQualif)

references Qualification;

alter table Pro\_tel add constraint EQU\_P\_t\_tel\_1

foreign key (tel)

references tel\_1;

alter table Pro\_tel add constraint EQU\_P\_t\_Propr\_FK

foreign key (id)

references Proprietaire;

alter table Proprietaire add constraint ID\_Proprietaire\_CHK

check(exists(select \* from Pro\_tel

where Pro\_tel.id = id));

alter table Stock add constraint REF\_Stock\_Stock\_FK

foreign key (codeStockage)

references Stockage;

alter table Stock add constraint REF\_Stock\_Piece

foreign key (Id)

references Piece;

alter table tel add constraint EQU\_tel\_Ouvri

foreign key (matricule)

references Ouvrier;

alter table tel\_1 add constraint ID\_tel\_1\_CHK

check(exists(select \* from Pro\_tel

where Pro\_tel.tel = tel));

alter table TradLibPiece add constraint REF\_TradL\_Piece\_FK

foreign key (Id)

references Piece;

alter table TradLibPiece add constraint REF\_TradL\_Langu\_FK

foreign key (CodeLangue)

references Langue;

alter table TypeOperation add constraint REF\_TypeO\_Langu\_FK

foreign key (CodeLangue)

references Langue;

alter table TypeOpOuv add constraint REF\_TypeO\_TypeO\_FK

foreign key (Code)

references TypeOperation;

alter table TypeOpOuv add constraint REF\_TypeO\_Ouvri

foreign key (matricule)

references Ouvrier;

alter table Vehicule add constraint REF\_Vehic\_Model\_FK

foreign key (idModel)

references Modele;

alter table Vehicule add constraint REF\_Vehic\_Conce\_FK

foreign key (idConcessionnaire)

references Concessionnaire;

alter table Visite add constraint REF\_Visit\_appar\_FK

foreign key (IdPropriete)

references appartenance;

-- Index Section

-- \_\_\_\_\_\_\_\_\_\_\_\_\_

create unique index ID\_appartenance\_IND

on appartenance (IdPropriete);

create index REF\_appar\_Vehic\_IND

on appartenance (idVehicule);

create index REF\_appar\_Propr\_IND

on appartenance (id);

create unique index ID\_Concessionnaire\_IND

on Concessionnaire (idConcessionnaire);

create unique index ID\_Consommation\_IND

on Consommation (IdConso);

create index REF\_Conso\_Piece\_IND

on Consommation (Id);

create index REF\_Conso\_Opera\_IND

on Consommation (IdOperation);

create unique index ID\_Devis\_IND

on Devis (Numero);

create unique index SID\_Devis\_Visit\_IND

on Devis (IdVisite);

create index REF\_Devis\_appar\_IND

on Devis (IdPropriete);

create unique index ID\_FactureImpaye\_IND

on FactureImpaye (idFacture);

create unique index SID\_Factu\_Visit\_IND

on FactureImpaye (IdVisite);

create unique index ID\_Langue\_IND

on Langue (CodeLangue);

create unique index SID\_Langue\_IND

on Langue (CodeLangue);

create index REF\_Langu\_Propr\_IND

on Langue (id);

create unique index ID\_Ligne\_IND

on Ligne (idLigne);

create index EQU\_Ligne\_Devis\_IND

on Ligne (Numero);

create unique index ID\_Ligne\_Ligne\_2\_IND

on LigneCommentaire (idLigne);

create index REF\_Ligne\_Piece\_IND

on LigneConsommation (Id);

create unique index ID\_Ligne\_Ligne\_1\_IND

on LigneConsommation (idLigne);

create index REF\_Ligne\_Opera\_IND

on LigneOperation (IdOperation);

create unique index ID\_Ligne\_Ligne\_IND

on LigneOperation (idLigne);

create unique index ID\_Marque\_IND

on Marque (CodeMarque);

create unique index ID\_Modele\_IND

on Modele (idModel);

create index REF\_Model\_Marqu\_IND

on Modele (CodeMarque);

create unique index ID\_Option\_IND

on Option (idOption);

create unique index ID\_optionNonSatandards\_IND

on optionNonSatandards (idVehicule, idOption);

create index REF\_optio\_Optio\_1\_IND

on optionNonSatandards (idOption);

create unique index ID\_optionStandards\_IND

on optionStandards (idModel, idOption);

create index REF\_optio\_Optio\_IND

on optionStandards (idOption);

create unique index ID\_Operation\_IND

on Operation (IdOperation);

create index REF\_Opera\_TypeO\_IND

on Operation (Code);

create index REF\_Opera\_Visit\_IND

on Operation (IdVisite);

create unique index ID\_Ouvrier\_IND

on Ouvrier (matricule);

create index REF\_Ouvri\_Quali\_IND

on Ouvrier (CodeQualif);

create unique index ID\_Piece\_IND

on Piece (Id);

create unique index ID\_Pro\_tel\_IND

on Pro\_tel (tel, id);

create index EQU\_P\_t\_Propr\_IND

on Pro\_tel (id);

create unique index ID\_Proprietaire\_IND

on Proprietaire (id);

create unique index ID\_Qualification\_IND

on Qualification (CodeQualif);

create unique index ID\_Stock\_IND

on Stock (Id, codeStockage);

create index REF\_Stock\_Stock\_IND

on Stock (codeStockage);

create unique index ID\_Stockage\_IND

on Stockage (codeStockage);

create unique index ID\_tel\_IND

on tel (matricule, tel);

create unique index ID\_tel\_1\_IND

on tel\_1 (tel);

create unique index ID\_TradLibPiece\_IND

on TradLibPiece (IdTrad);

create index REF\_TradL\_Piece\_IND

on TradLibPiece (Id);

create index REF\_TradL\_Langu\_IND

on TradLibPiece (CodeLangue);

create unique index ID\_TypeOperation\_IND

on TypeOperation (Code);

create index REF\_TypeO\_Langu\_IND

on TypeOperation (CodeLangue);

create unique index ID\_TypeOpOuv\_IND

on TypeOpOuv (matricule, Code);

create index REF\_TypeO\_TypeO\_IND

on TypeOpOuv (Code);

create unique index ID\_Vehicule\_IND

on Vehicule (idVehicule);

create unique index SID\_Vehicule\_IND

on Vehicule (NumeroChassis);

create index REF\_Vehic\_Model\_IND

on Vehicule (idModel);

create index REF\_Vehic\_Conce\_IND

on Vehicule (idConcessionnaire);

create unique index ID\_Visite\_IND

on Visite (IdVisite);

create unique index SID\_Visite\_IND

on Visite (n°Visite, dateEntree);

create index REF\_Visit\_appar\_IND

on Visite (IdPropriete);