



UNIVERSITÉ  
DE LORRAINE



Saint-Dié-des-Vosges

# **S2.04: Exploitation d'une base de données**

## **Sommaire:**

- 1) Introduction**
- 2) Rappel du sujet**
- 3) Analyse et recherche**
- 4) Modèle E/A**
- 5) Modèle relationnel**
- 6) Alimentation de la base de données**
- 7) Déclencheurs**
- 8) Méthodes d'analyse**

## Introduction

Dans le cadre de cette SAÉ 2.04: Exploitation d'une base de données, il a fallu créer à partir de 0 une base de données qui couvre les compétences que nous avons acquis durant ce semestre. Cette base de données devait être un minimum complète pour pouvoir être intéressante et pertinente à analyse, mais aussi contenir des variables quantitative pour la visualisation en rapport avec les apprentissages mathématiques, mais aussi contenir des déclencheurs afin de faciliter l'exploitation de la base de données, afin de simuler une utilisation dans le monde professionnel.

## Rappel du sujet

1. Faites une Analyse/recherche pour identifier les entités, les attributs et les associations (identifier d'éventuelles nouvelles entités).
2. Donner le modèle E/A de la base de données en indiquant les associations et les cardinalités (ne pas utiliser Workbench ici, mais un logiciel de dessin pour dessiner le modèle E/A...).
3. Créer le modèle relationnel de cette base de données en précisant à chaque fois la règle de passage du MCD au MLD que vous appliquez. On donne ici le Schéma sur Workbench et on utilise le « forward engineering » pour générer le code SQL.
4. Alimentation de la base de données : Insérer des données dans les tables.
5. Créer des déclencheurs de votre choix qui vous aident à gérer votre base de données (Par exemple le déclencheur « update\_stock\_after\_sale » pour mettre à jour le niveau de stock du produit concerne après chaque vente).
6. Appliquer différentes méthodes d'analyse exploratoires mathématiques sur les différents attributs de la base de données. Vous pouvez faire des calculs de moyenne, de variance, d'écart type, de minimum, de maximum, Etendu, faire des Boxplot, des histogrammes, des diagrammes en bâton (des Barplot), ....

## Analyse et recherche

Le contexte retenu est donc une chaîne de concessions automobiles. Il a donc fallut déterminer les entités qui permettent de gérer cette base de données. Ces entités ne gèrent que la vente de véhicules sans promotion, elles ne permettent pas la gestion des achats de véhicules, ni des reprises.

### Entités:

- Concession
- Personne
- Employé
- Client
- Stock
- Véhicule
- Fiche\_Technique
- Marque
- Modèle
- Opération

### Attributs:

#### **Concession**

Concession\_Id: int (Clé primaire)  
Ville: varchar(45)  
Adresse: varchar(45)

#### **Personne**

Id\_Personne: int (Clé primaire)  
Nom: varchar(45)  
Prénom: varchar (45)  
Adresse: varchar(45)  
Nb\_Téléphone: varchar(45)  
Email: varchar(45)

#### **Employé**

Id\_Employé: int (Clé primaire)  
Id\_Personne: int (Clé étrangère vers la table Personne)  
Id\_Concession: int (Clé étrangère vers la table Concession)  
Poste: varchar(45)

**Client**

Id\_Client: int (Clé primaire)  
Id\_Personne: int (Clé étrangère vers la table Personne)  
Type\_Client: varchar(45)

**Stock**

Id\_Stock: int (Clé primaire)  
Id\_Concession: int (Clé étrangère vers la table Concession)  
Nombre\_Véhicules

**Véhicule**

Id\_Véhicule: int (Clé primaire)  
Id\_Stock: (Clé étrangère vers la table Stock)  
Id\_Modèle: int (Clé étrangère vers la table Modèle)  
Génération: varchar(45)  
Marque: varchar(45) (Clé étrangère vers la table Marque)  
Id\_Fiche\_Technique: int (Clé étrangère vers la table Fiche\_Technique)  
Couleur: varchar(45)  
Finition: varchar(45)  
État: varchar(45)  
Kilométrage: int  
Provenance: varchar(45)  
Prix: float(8,2)

**Fiche\_Technique**

Id\_Fiche\_Technique: int (Clé primaire)  
Bloc\_Moteur: varchar(45)  
Puissance\_DIN: int  
Couple: int  
Énergie: varchar(45)  
Boite\_De\_Vitesse: varchar(45)  
ConsommationAu100: float  
CritAir: varchar(10)

**Marque**

Marque: varchar(45) (Clé primaire)  
Origine: varchar(45)

**Modèle**

Id\_Modèle: int (Clé primaire)  
Marque: varchar(45) (Clé étrangère vers la table Marque)  
Nom: varchar(45)

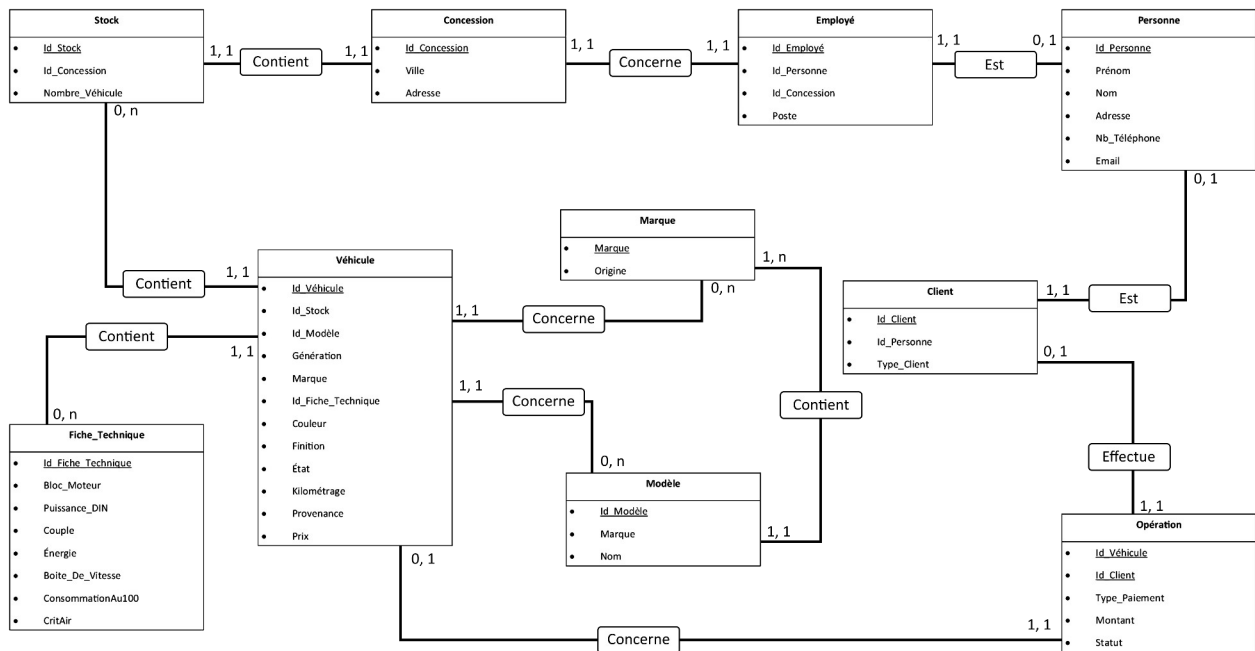
## Opération

Type\_Paiement: varchar(45)  
Montant: float  
Statut: varchar(45)  
Id\_Véhicule: int (Composante clé primaire)  
Id\_Client: int (Composante clé primaire)

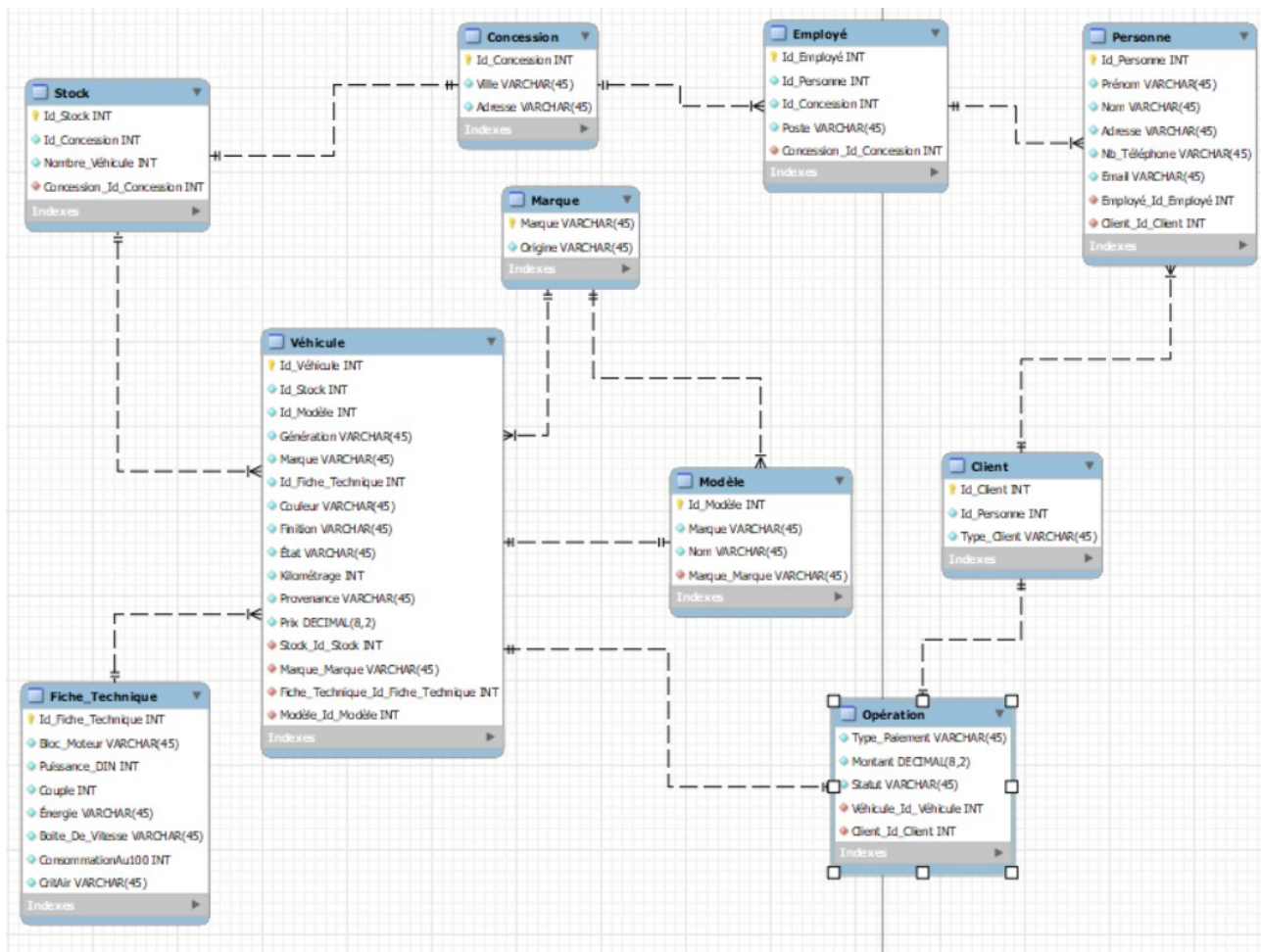
## Associations

- 1 **Concession** possède 1 **Stock**
- 1 **Concession** concerne 1, n **Employé**
  
- 1 **Personne** peut être 1 **Employé**
- 1 **Personne** peut être 1 **Client**
  
- 1 **Employé** est une **Personne**
- 1 **Employé** est concerné par 1 **Concession**
  
- 1 **Client** est une **Personne**
- 1 **Client** est concerné par 0, n **Opération**
  
- 1 **Stock** est possédé par 1 **Concession**
- 1 **Stock** possède n **Véhicules**
  
- 1 **Véhicule** est possédé par 1 **Stock**
- 1 **Véhicule** concerne 1 **Modèle**
- 1 **Véhicule** concerne 1 **Marque**
- 1 **Véhicule** possède 1 **Fiche\_Technique**
- 1 **Véhicule** est concerné par 0, 1 **Opération**
  
- 1 **Fiche\_Technique** est possédé par 0, n **Véhicule**
  
- 1 **Marque** est concerné par 0,n **Véhicule**
- 1 **Marque** contient 1,n **Modèle**
  
- 1 **Modèle** est concerné par 1 **Marque**
- 1 **Modèle** est concerné par 0,n **Véhicule**
  
- 1 **Opération** concerne 1 **Véhicule**
- 1 **Opération** concerne 1 **Client**

# Modèle E/A



# Modèle relationnel





## Alimentation de la base de données

Des données ont été ajoutées à la base de données. Une partie a pu être générée par intelligence artificielle mais une autre partie a dû être faite à la main car l'intelligence artificielle n'était pas en capacité de rechercher les informations demandées:

- Concession: 5 concessions
- Personne: 60 Personnes
- Employé: 40 Employés
- Client: 20 Clients
- Stock: 5 Stocks
- Véhicule: 300 Véhicules
- Fiche\_Technique: 74 Fiches Technique
- Marque: 18 Marques
- Modèle: 60 Modèles
- Opération: 18 Opérations

## Déclencheurs

Nous avons mis en place plusieurs déclencheurs afin de faciliter la gestion de la base de données.

### *Mise à jour du Stock*

Dès qu'un véhicule est supprimé ou ajouté, le nombre de véhicule du stock correspondant au véhicule est mis à jour. La mise à jour fonctionne si plusieurs véhicules sont supprimé ou ajouté en même temps. Deux trigger ont été fait: un pour l'ajout, un pour la suppression

```
DELIMITER //  
CREATE TRIGGER `after_vehicle_delete` AFTER DELETE ON `véhicule`  
FOR EACH ROW BEGIN  
    DECLARE quantite INT;  
    SET quantite = (SELECT COUNT(*) FROM Véhicule WHERE Id_Stock = OLD.Id_Stock);  
    UPDATE stock  
    SET Nombre_Véhicule = quantite  
    WHERE Id_Stock = OLD.Id_Stock;  
END //  
DELIMITER ;
```

```
DELIMITER //  
CREATE TRIGGER `before_vehicle_insert` AFTER INSERT ON `véhicule`  
FOR EACH ROW BEGIN  
    DECLARE quantite INT;  
    SET quantite = (SELECT COUNT(*) FROM Véhicule WHERE Id_Stock = NEW.Id_Stock);  
    UPDATE stock  
    SET Nombre_Véhicule = quantite  
    WHERE Id_Stock = NEW.Id_Stock;  
END //  
DELIMITER ;
```

### ***Ajustements du prix***

Un trigger a été ajouté, qui permet d'ajuster le prix si on augmente le kilométrage d'un véhicule en stock. Son prix augmentera avec un rapport de 100 euros pour 1 000 kilomètres. En revanche, cette mise à jour ne sera pas effectué si le prix final du véhicule passera en dessous de 5 000 euros. Aucun véhicule ne doit avoir de prix inférieur à 5 000 euros sauf pour les véhicules ajouté manuellement.

**DELIMITER //**

**CREATE TRIGGER** ajustement\_prix\_km **BEFORE UPDATE ON** Véhicule  
**FOR EACH ROW BEGIN**

**DECLARE** kilometer\_diff INT;  
**DECLARE** price\_diff DECIMAL(10, 2);  
**DECLARE** new\_prix DECIMAL(10, 2);

**IF OLD.Kilométrage < NEW.Kilométrage THEN**  
**SET** kilometer\_diff = **NEW.Kilométrage** - **OLD.Kilométrage**;  
**SET** price\_diff = kilometer\_diff / 1000 \* 100;  
**SET** new\_prix = **OLD.Prix** - price\_diff;

**IF** new\_prix < 5000 **THEN**  
**SET NEW.Prix** = 5000;  
**ELSE**  
**SET NEW.Prix** = **OLD.Prix** - price\_diff;  
**END IF**;

**END IF**;  
**END //**  
**DELIMITER ;**

***Suppression d'un véhicule concerné par une opération***

Si on veut supprimer un véhicule de la table Véhicule, mais qu'il est concerné par une opération, la requête échouera. Un simple trigger a donc été implémenté, qui permet d'avant d'essayer de supprimer le véhicule de la table Véhicule, va d'abord le supprimer dans la table Opération.

```
DELIMITER //  
CREATE TRIGGER delete_vehicle BEFORE DELETE ON Véhicule  
FOR EACH ROW BEGIN  
    DELETE FROM Opération WHERE Véhicule_ID_Véhicule = OLD.Id_Véhicule;  
END //  
DELIMITER ;
```

## Méthodes d'analyse

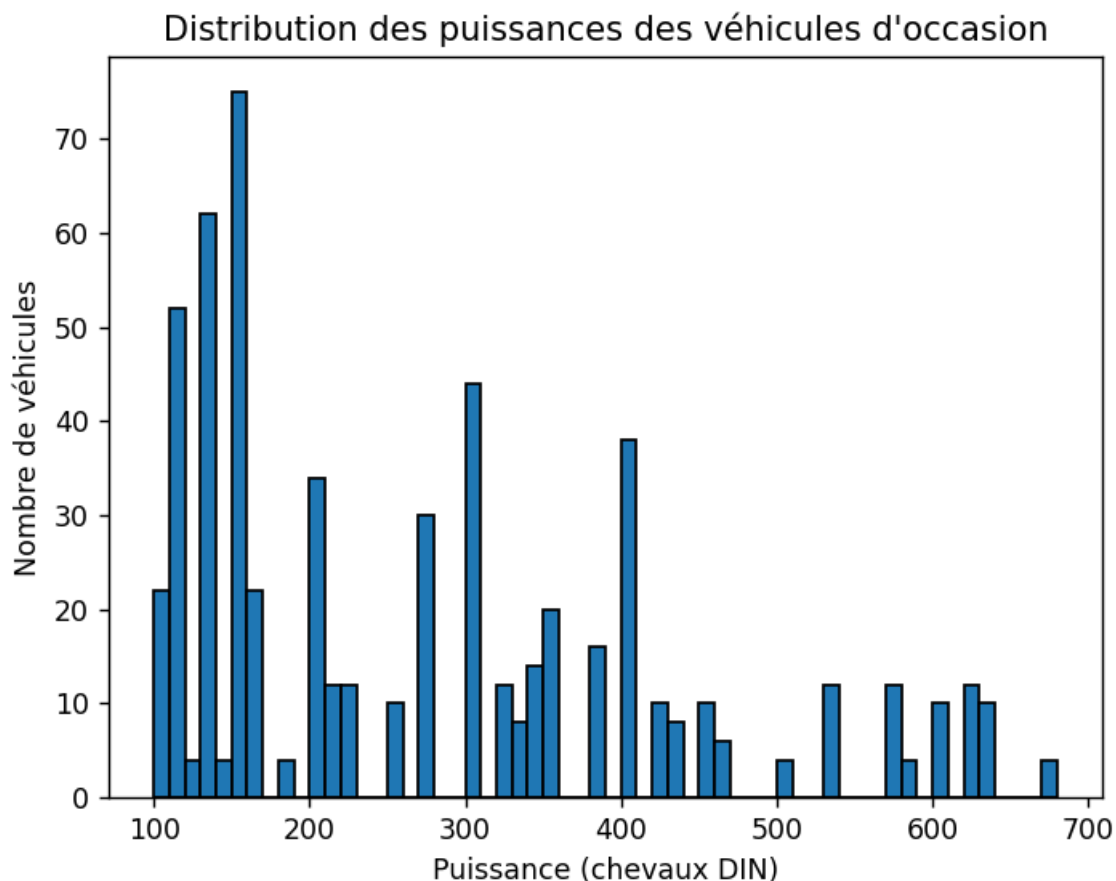
L'énoncé demandait 3 analyses pour la visualisation des données:

- Une analyse avec une variable quantitative discrète
- Une analyse avec une variable quantitative continue
- Une analyse avec deux variable quantitatives discrètes

### ***Analyse avec une variable quantitative discrète***

Nous avons ici choisit comme variable d'analyse la puissance des véhicules. Ainsi, nous avons en X les valeurs qui représentent la puissance, et en Y la quantité de véhicule ce qui nous permet d'obtenir ce graphe en bâton.

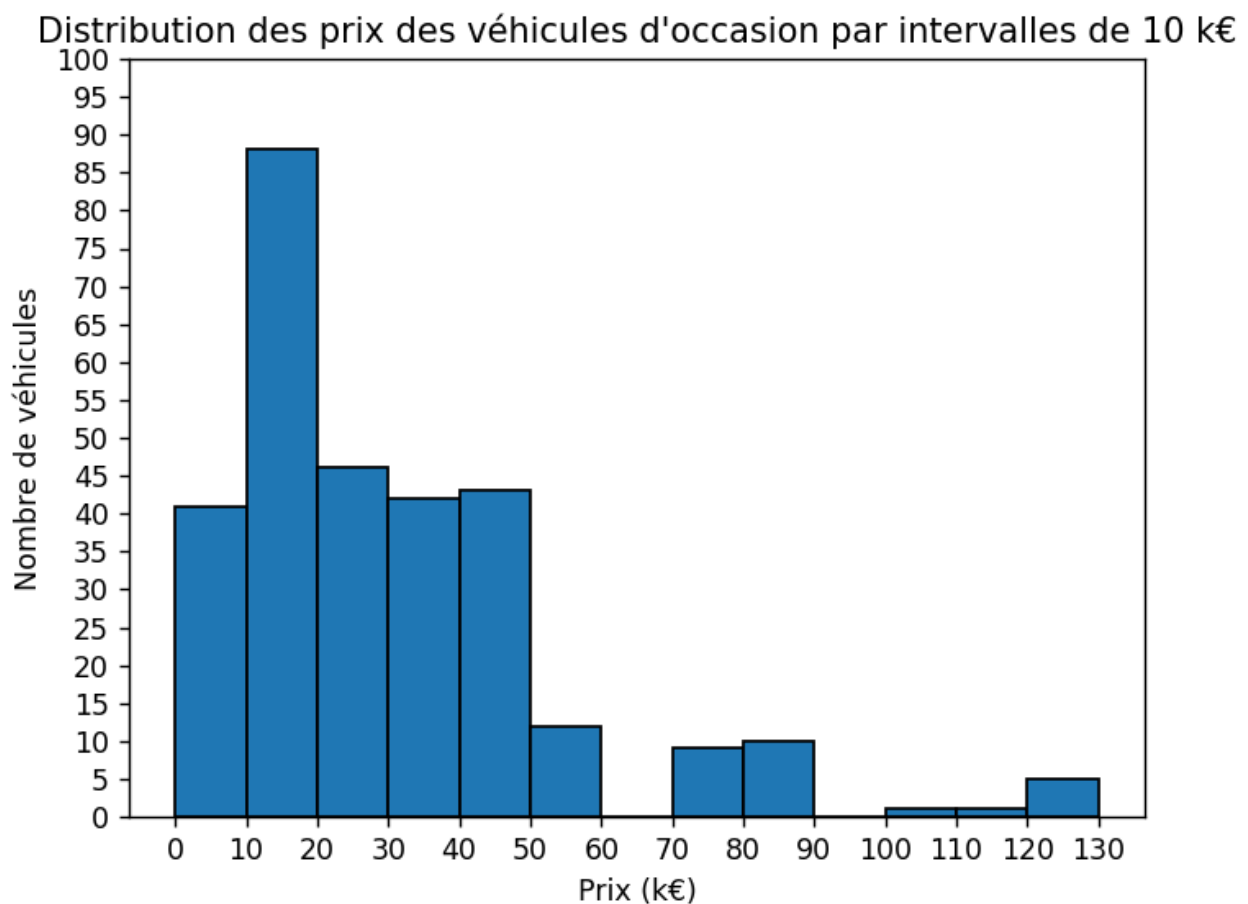
Le code a été rédigé sur Python car le logiciel nous est plus familier et car des travaux de visualisation ont été effectués dessus en TP



**Analyse avec une variable quantitative continue**

Pour cette analyse, nous avons du choisir une variable continue, donc sous forme d'intervalle pour l'axe X. La variable retenue a donc été le prix où il a été analysé sous forme d'intervalle. Les intervalles sont régulier et d'une taille de 10 sachant qu'ils représentent le prix en millier d'euros.

Le code a ici aussi été fait sur Python et permet d'obtenir ce rendu



### **Analyse avec deux variables quantitatives discrètes**

Nous avons du choisir deux variables singulières pour et les traiter, l'une pour X et l'autre pour Y. Nous avons choisit dans la table Fiche\_Technique la puissance pour l'axe X et la consommation pour l'axe Y car ce sont des variables qui ne représentent pas un intervalle, et qui sont numérique ce qui permet l'analyse quantitative.

Toujours sur Python, nous avons rédigé un code qui permet d'obtenir un nuage de point avec une courbe qui les relie afin d'observer le niveau de consommation selon la puissance d'un véhicule.

