

# Conception d'une base de données pour un Garage

---

A.U : 2022/2023

**MODULE : INFO0304**

Réalisé par : Mohamed MOHAMED et AHNOUDJ Lina Laaldja

12/7/2022



## Table of Contents

|   |    |
|---|----|
| Réalisé par : Mohamed MOHAMED et AHNOUDJ Lina Laaldja .....                     | 0  |
| I. Introduction.....  | 3  |
| II. Analyse .....   | 4  |
| Matrice de flux.....  | 4  |
| 1. Cahier de charges .....  | 4  |
| 2. Dictionnaire de données.....   | 5  |
| a) Les entités .....  | 5  |
| b) Les associations .....   | 7  |
| 3. Le modèle conceptuel des données MCD.....                                    | 11 |
| 4. Le modèle logique des données MLD .....                                      | 12 |
| 5. Le modèle relationnelle (MLD textuelle) :.....                               | 13 |
| 6. Normalisation.....   | 13 |
| 7. Modèle conceptuel de traitement (MCT).....                                   | 14 |
| III. Partie SQL et Algèbre relationnelle .....                                  | 16 |
| R1 : La liste des clients avec leurs véhicules .....                            | 16 |
| SQL :.....  | 16 |
| Algèbre Relationnelle : .....   | 17 |
| R2 : La liste des rendez-vous clients de la journée .....                       | 17 |
| SQL :.....  | 17 |
| Algèbre Relationnelle : .....   | 17 |
| R3 : La liste des rendez-vous clients de la semaine.....                        | 18 |
| SQL :.....  | 18 |
| Algèbre Relationnelle : .....   | 18 |
| R4 : Le planning des mécaniciens .....  | 19 |
| SQL :.....  | 19 |
| Algèbre Relationnelle : .....   | 19 |
| R5 : La facturation d'une réparation.....                                       | 20 |
| SQL :.....  | 20 |
| Algèbre Relationnelle : .....   | 20 |
| R6 : La liste des réparations faites sur un véhicule depuis un temps donné..... | 21 |
| SQL :.....  | 21 |
| Algèbre Relationnelle : .....   | 21 |
| R7 : Le montant total de la facturation d'une réparation .....                  | 22 |
| SQL :.....  | 22 |
| Algèbre Relationnelle : .....   | 22 |
| R8 : Liste des réparations d'entretien d'un véhicule .....                      | 23 |

|                               |    |
|-------------------------------|----|
| SQL : .....                   | 23 |
| Algèbre Relationnelle : ..... | 23 |
| Conclusion .....              | 24 |
| Annexe .....                  | 25 |
| Webographie .....             | 26 |

# I. Introduction

Ce projet a pour but d'aider le grand garage Medlin à bien gérer ses clients, le suivi des réparations ainsi que le planning des mécaniciens à l'aide d'une base de données relationnelle. Pour cette fin, nous avons été contactés afin de créer cette base de données qui permet de stocker de gros volumes d'informations et partager celles-ci à une communauté de personnes (clients, techniciens et autre employés). De nos jours beaucoup de secteurs utilisent des systèmes de gestion des bases de données pour faciliter la gestion générale de leurs entreprises.

En particulier, la complexité d'un garage qui reçoit un très grand nombre de voitures par mois rendent l'utilisation des bases de données très utile pour mettre en forme ce type des relations et facilité la gestion.

Nous avons dû observer longuement le garage avant de créer notre base de données. Nous avons dû relever tous les employés ainsi qu'examiner la structure du garage.

Nous avons pu noter que le garage se compose d'une grande salle, où se font les réparations ainsi qu'un parking. De plus, le garage dispose d'un ensemble des mécaniciens et d'un ensemble des secrétaires et un chef de service. La bases de données a été conçue selon la méthode Merise, en établissant tout d'abord un dictionnaire de données, puis un Modèle Conceptuel de Données (MCD) et Modèle Relationnel (MR), pour aboutir finalement au Modèle Physique de Données (MPD) programmé en SQL avec MySQL.

Ce projet comporte trois parties :

- Une partie analyse : comprenant un MCD et MLD et qui explique les choix de développement de la base de données;
- Une partie SQL et Algèbre relationnelle : comprenant entre autres les requêtes basiques d'insertion et de suppression de données pour chacune des tables, le script SQL permettant la création de la base et l'insertion d'un certain volume de données, ainsi que les requête en SQL et Algèbre relationnelle;

➤ **Nom garage** : Garage Medlin

➤ **Logo** :



## II. Analyse

Dans cette section, nous allons définir un nombre de points concernant la partie analyse de projet. Nous allons traiter la partie cahier de charge et ensuite faire les parties modélisations MCD et MLD.

### Matrice de flux

Une matrice de flux nous permet de structurer les échanges entre nos différents acteurs. Notre domaine d'étude étant un garage. Nous avons pu déterminer 3 acteurs : Client, Secrétaire, et Mécanicien. Par exemple on peut voir les échanges d'un client vis à vis d'un secrétaire : Un client consulte un secrétaire et reçoit une confirmation d'un rendez-vous de sa part.

|            | Client                                    | Secrétaire   | Mécanicien              |
|------------|---|--|-------------------------|
| Client     |   | ~Etre Reçu<br>~Effectuer une facturation<br>~confirmer rendez-vous | ~Recevoir un réparation |
| Secrétaire | ~Recevoir<br>~Enregistrer une facturation |  | ~Attribuer des tâches   |
| Mécanicien | ~réparer véhicule                         | ~Consulter le planning tâches                                      |                         |

### 1. Cahier de charges

Pour réaliser ce cahier de charge nous avons lu méticuleusement l'énoncé de sujet du projet et lu les informations sur les sites [1], [2], [3], [4] et [5] pour bien comprendre et retirer les informations utiles à la conception de la base de données d'un garage mécaniciens. Nous avons constaté qu'il y a des charges gérable par la modélisation Merise Entité et Association, et qu'il y a d'autre faisable à travers des requêtes SQL ou Algèbre relationnelle. Les charges qu'on peut modéliser sont décrites dans le paragraphe ci-dessous.

Le garage est conçu pour recevoir des véhicules des clients ayant besoin d'être traité par un mécanicien. Chaque **véhicule** est décrit par sa **marque**, son **type**, son numéro d'immatriculation et sa date de première mise en circulation. Un **client** est caractérisé par son nom, son prénom, son adresse, email et numéro de téléphone. Sur chaque voiture on effectue un ensemble de **réparations** qui est décrit par une date, kilométrage à cette date et le nombre des heures passé dans la réparation. Une réparation peut être constituée d'une ou plusieurs **opérations**. Chaque réparation donne lieu à la facturation de la main d'œuvre, des **pièces** et de la **prestation**. Les réparations sont effectuées par les mécaniciens salariés du garage. Chaque **mécanicien** est caractérisé par son nom, son prénom, son adresse, email et numéro de téléphone. Les clients doivent prendre **rendez-vous** avant de venir au garage.

### Salles et personnel :

| Nombre salles |            | Nombre techniciens | Nombre Secrétaires |
|---------------|------------|--------------------|--------------------|
| Réparation    | D'attentes | 6                  | 1                  |
| 1             | 1          |                    |                    |

Charges gérables avec les requêtes :

- La liste des rendez-vous clients de la journée ou de la semaine,
- Le planning des mécaniciens,
- La facturation de la réparation,
- La liste des réparations faites sur un véhicule depuis un temps donné,
- ...etc.

## 2. Dictionnaire de données

Dans cette sous partie, nous allons préciser les détails des entités ainsi que les associations et relations entre entités de notre modèle MCD. La conception de ce modèle est précédée par la réalisation d'un dictionnaire de données après avoir trouvé les entités et les associations entre eux.

### a) Les entités

- Chaque **véhicule** est décrit par sa **marque**, son **type**, son numéro d'immatriculation, sa date de première mise en circulation et un kilométrage.

L'entité **Véhicule** :

| Attribut         | Signification                      | Type/Taille | Exemple    |
|------------------|------------------------------------|-------------|------------|
| Id_Vehicule      | Identifiant du véhicule            | INT         | 1          |
| Date_circulation | Date début circulation du véhicule | DATE        | 11/11/2010 |

L'entité **Type**:

| Attribut | Signification       | Type/Taille | Exemple |
|----------|---------------------|-------------|---------|
| Id_Type  | Identifiant du type | INT         | 1       |
| Titre    | Nom du type         | VARCHAR(20) | 308     |

L'entité **Marque** :

| Attribut  | Signification            | Type/Taille | Exemple |
|-----------|--------------------------|-------------|---------|
| Id_Marque | Identifiant de la marque | INT         | 1       |
| Nom       | Nom de la marque         | VARCHAR(20) | Renault |

- Un **client/mécanicien** est caractérisé par son nom, son prénom, son adresse, email et numéro de téléphone.

L'entité **Client/Mécanicien** :

| Attribut  | Signification         | Type/Taille | Exemple          |
|-----------|-----------------------|-------------|------------------|
| Id_Client | Identifiant du client | INT         | 1                |
| Nom       | Nom du client         | VARCHAR(20) | Mohamed          |
| Prénom    | Prénom du client      | VARCHAR(20) | Mahmoud          |
| Adresse   | Adresse du client     | VARCHAR(65) | Reims, Rue 2, 25 |
| Télé      | Téléphone du client   | VARCHAR(20) | 0694564789       |
| Email     | Email du client       | VARCHAR(30) | mohamed@mail.com |

| Attribut       | Signification              | Type/Taille   | Exemple          |
|----------------|----------------------------|---------------|------------------|
| Id_Mecaniciene | Identifiant du mécanicien  | INT           | 1                |
| Nom            | Nom du mécanicien          | VARCHAR(20)   | Mohamed          |
| Prénom         | Prénom du mécanicien       | VARCHAR(20)   | Mahmoud          |
| Adresse        | Adresse du mécanicien      | VARCHAR(65)   | Reims, Rue 2, 25 |
| Email          | Email du mécanicien        | VARCHAR(30)   | mohamed@mail.com |
| Télé           | Téléphone du mécanicien    | VARCHAR(20)   | 0694564789       |
| Prix_heure     | Main d'œuvre du mécanicien | DECIMAL(10,2) | 50               |

- **Réparations** est décrit par une date, kilométrage à cette date et le nombre des heures passé dans la réparation.

L'entité **Réparation** :

| Attribut        | Signification                 | Type/Taille | Exemple    |
|-----------------|-------------------------------|-------------|------------|
| Id_Reparation   | Identifiant de la réparation  | INT         | 1          |
| Date_reparation | La date de la réparation      | DATE        | 11/22/2022 |
| Kilométrage     | Kilométrage à la réparation   | DATE        | 12/22/2022 |
| Nbr_heure       | Nombre heure de la réparation | INT         | 6          |

- Une réparation peut être constituée d'une ou plusieurs **opérations**.

L'entité **Opération** :

| Attribut     | Signification           | Type/Taille   | Exemple |
|--------------|-------------------------|---------------|---------|
| Id_operation | Identifiant d'opération | INT           | 1       |
| Titre        | Le nom d'opération      | VARCHAR(25)   | Vidange |
| Prix         | Le prix d'opération     | DECIMAL(10,2) | 60.0    |
| TVA          | La TVA d'opération      | DECIMAL(6,2)  | 20.0    |

- Chaque réparation donne lieu à la facturation de la main d'œuvre, des **pièces** et de la **prestation**.

L'entité **Pièce/Prestation** : (même caractéristiques)

| Attribut | Signification           | Type/Taille   | Exemple     |
|----------|-------------------------|---------------|-------------|
| Id_piece | Identifiant de la pièce | INT           | 1           |
| Titre    | Le nom de la pièce      | VARCHAR(25)   | Compresseur |
| Prix     | Le prix de la pièce     | DECIMAL(10,2) | 60.0        |
| TVA      | La TVA de la pièce      | DECIMAL(6,2)  | 20.0        |

- Les clients doivent prendre **rendez-vous** avant de venir au garage.

L'entité **Rendez-vous**:

| Attribut         | Signification              | Type/Taille | Exemple    |
|------------------|----------------------------|-------------|------------|
| Id_rendez_vous   | Identifiant du rendez-vous | NUMBER(10)  | 1          |
| Date_rendez_vous | Le nom du rendez-vous      | DATE        | 12/05/2022 |

## b) Les associations

- L'association un à plusieurs **Posséder**:  
 Cardinalité 1.1 : Un véhicule appartient à un client.  
 Cardinalité 1.n : Un client possède un ou plusieurs véhicules.

| Entité 1 | Entité 2 | Nom en MCD | Nom en MLD/SQL |
|----------|----------|------------|----------------|
| Véhicule | Client   | Posséder   |                |

- L'association un à plusieurs **Prendre** :  
 Cardinalité 1.1 : Un rendez-vous concerne un client.  
 Cardinalité 1.n : Un client peut prendre un ou plusieurs rendez-vous.

| Entité 1 | Entité 2    | Nom en MCD | Nom en MLD/SQL |
|----------|-------------|------------|----------------|
| Client   | Rendez-vous | Prendre    |                |

- L'association un à plusieurs **Correspondre**:  
 Cardinalité 1.1 : Un rendez-vous concerne un et un seul véhicule.  
 Cardinalité 1.n : Un véhicule peut être l'objet d'un ou plusieurs rendez-vous.

| Entité 1    | Entité 2 | Nom en MCD   | Nom en MLD/SQL |
|-------------|----------|--------------|----------------|
| Rendez-vous | Véhicule | Correspondre |                |

- L'association un à plusieurs **fabriquer**:  
 Cardinalité 1.1 : Un véhicule est fabriqué par une seule marque.  
 Cardinalité 1.n : Une marque concerne un ou plusieurs véhicules.

| Entité 1 | Entité 2 | Nom en MCD | Nom en MLD/SQL |
|----------|----------|------------|----------------|
| Marque   | Véhicule | Fabriquer  |                |

- L'association un à plusieurs **Avoir**:  
 Cardinalité 1.1 : Un véhicule correspond à un et un seul type.  
 Cardinalité 1.n : Un type correspond à un ou plusieurs véhicules.

| Entité 1 | Entité 2 | Nom en MCD | Nom en MLD/SQL |
|----------|----------|------------|----------------|
| Véhicule | Type     | Avoir      |                |



- L'association un à plusieurs **Concerner**:  
 Cardinalité 1.1 : Un rendez-vous concerne un seul véhicule.  
 Cardinalité 1.n : Un véhicule fait l'objet d'un ou plusieurs rendez-vous.

| Entité 1 | Entité 2    | Nom en MCD | Nom en MLD/SQL |
|----------|-------------|------------|----------------|
| Véhicule | Rendez-vous | Concerner  |                |

- L'association plusieurs à plusieurs **Faire**:  
 Cardinalité 1.n : Un mécanicien peut faire une ou plusieurs réparations.  
 Cardinalité 1.n : Une réparation est faite par un ou plusieurs mécaniciens.

| Entité 1   | Entité 2   | Nom en MCD | Nom en MLD/SQL |
|------------|------------|------------|----------------|
| Mécanicien | Réparation | Faire      | Traitement     |

Ce type d'association donne lieu à une relation dans le modèle MLD.

La relation **Traitement** :

| Attribut      | Signification                | Type/Taille | Exemple |
|---------------|------------------------------|-------------|---------|
| Id_reparation | Identifiant de la réparation | INT         | 1       |
| Id_mecanicien | Identifiant du mécanicien    | INT         | 2       |

- L'association plusieurs à plusieurs **attribuer**:  
 Cardinalité 1.n : Un mécanicien peut être attribué à une ou plusieurs réparations.  
 Cardinalité 1.n : Une réparation est attribuée par un ou plusieurs mécaniciens.

| Entité 1   | Entité 2   | Nom en MCD | Nom en MLD/SQL |
|------------|------------|------------|----------------|
| Mécanicien | Réparation | Attribuer  | Tache          |

Ce type d'association donne lieu à une relation dans le modèle MLD.

La relation **Tache** :

| Attribut      | Signification                | Type/Taille | Exemple |
|---------------|------------------------------|-------------|---------|
| Id_reparation | Identifiant de la réparation | INT         | 1       |
| Id_mecanicien | Identifiant du mécanicien    | INT         | 2       |

- L'association plusieurs à plusieurs **Utiliser**:  
 Cardinalité 1.n : Une pièce peut faire l'objet d'une ou plusieurs réparations.  
 Cardinalité 1.n : Une réparation peut inclure une ou plusieurs pièces.

| Entité 1 | Entité 2   | Nom en MCD | Nom en MLD/SQL |
|----------|------------|------------|----------------|
| Pièce    | Réparation | Utiliser   | Remplacement   |

Ce type d'association donne lieu à une relation dans le modèle MLD.

La relation **Remplacement** :

| Attribut      | Signification                | Type/Taille | Exemple |
|---------------|------------------------------|-------------|---------|
| Id_reparation | Identifiant de la réparation | INT         | 1       |
| Id_piece      | Identifiant de la pièce      | INT         | 2       |

- L'association plusieurs à plusieurs **Inclure**:  
 Cardinalité 1.n : Une opération est incluse dans une ou plusieurs réparations.  
 Cardinalité 1.n : Une réparation est faite par avec une ou plusieurs opérations.

| Entité 1  | Entité 2   | Nom en MCD | Nom en MLD/SQL       |
|-----------|------------|------------|----------------------|
| Opération | Réparation | Inclure    | Operation_reparation |

Ce type d'association donne lieu à une relation dans le modèle MLD.

La relation **Remplacement** :

| Attribut      | Signification                | Type/Taille | Exemple |
|---------------|------------------------------|-------------|---------|
| Id_reparation | Identifiant de la réparation | INT         | 1       |
| Id_operation  | Identifiant d'opération      | INT         | 2       |

- L'association plusieurs à plusieurs **Contenir**:  
 Cardinalité 1.n : Une prestation est contenue dans une ou plusieurs réparations.  
 Cardinalité 1.n : Une réparation est faite par avec une ou plusieurs prestations.

| Entité 1   | Entité 2   | Nom en MCD | Nom en MLD/SQL        |
|------------|------------|------------|-----------------------|
| Prestation | Réparation | Contenir   | Prestation_reparation |

Ce type d'association donne lieu à une relation dans le modèle MLD.

La relation **Prestation\_reparation** :

| Attribut      | Signification                | Type/Taille | Exemple |
|---------------|------------------------------|-------------|---------|
| Id_reparation | Identifiant de la réparation | INT         | 1       |
| Id_prestation | Identifiant de la prestation | INT         | 2       |

- L'association plusieurs à plusieurs **Programmer**:  
 Cardinalité 1.n : Une réparation peut être programmée pour un ou plusieurs véhicules.  
 Cardinalité 1.n : Un véhicule peut avoir une ou plusieurs réparations programmées.

| Entité 1 | Entité 2   | Nom en MCD | Nom en MLD/SQL         |
|----------|------------|------------|------------------------|
| Véhicule | Réparation | Programmer | reparation_d_entretien |

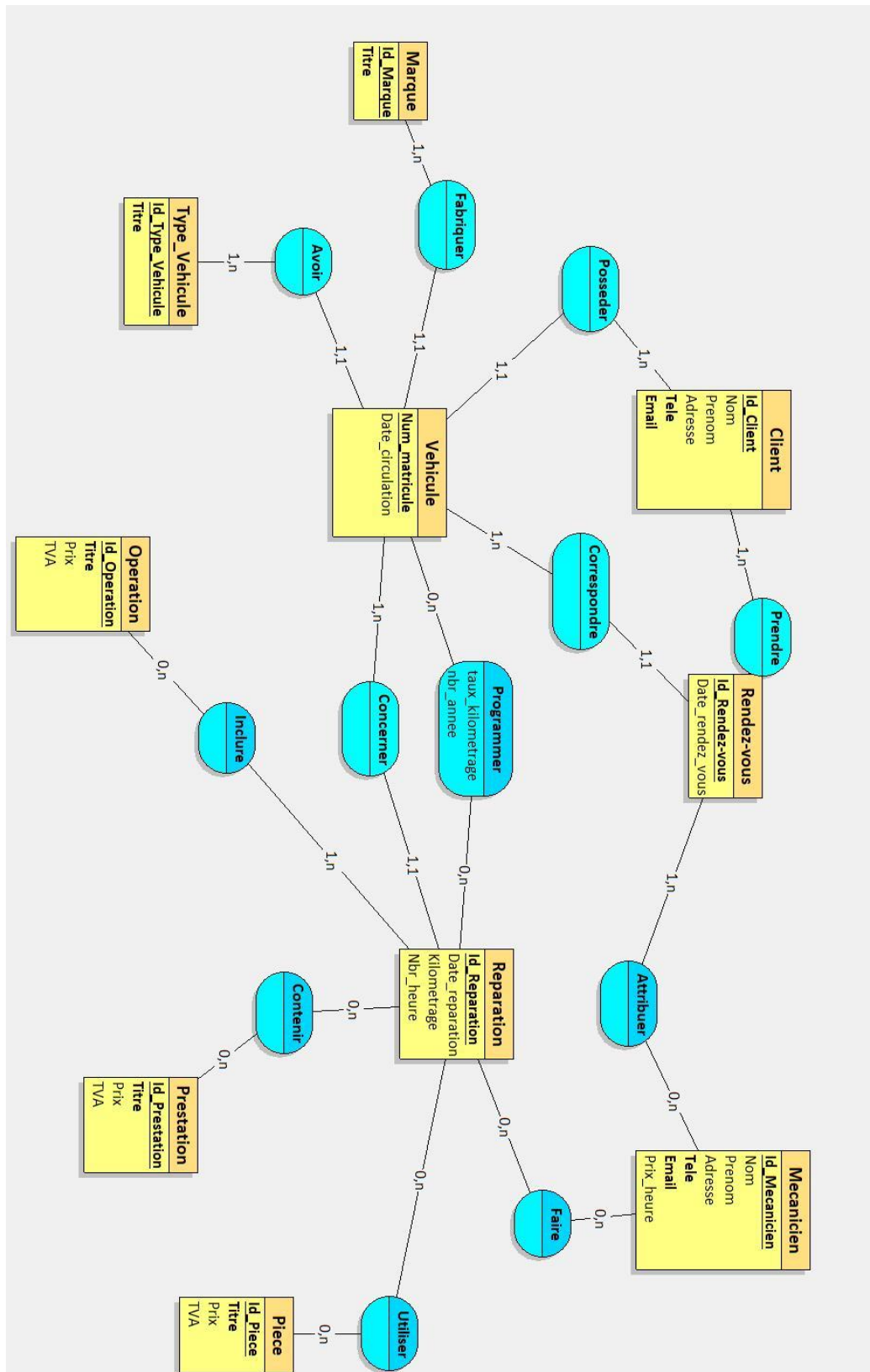
Ce type d'association donne lieu à une relation dans le modèle MLD.

La relation **Reparation\_d\_entretien**:

| Attribut         | Signification                | Type/Taille   | Exemple |
|------------------|------------------------------|---------------|---------|
| Id_reparation    | Identifiant de la réparation | INT           | 1       |
| Id_vehicule      | Identifiant du véhicule      | INT           | 2       |
| Taux_kilometrage | Date de vérification         | DECIMAL(15,2) | 12546.5 |
| Nbr_annee        | Résultat de vérification     | INT           | 4       |

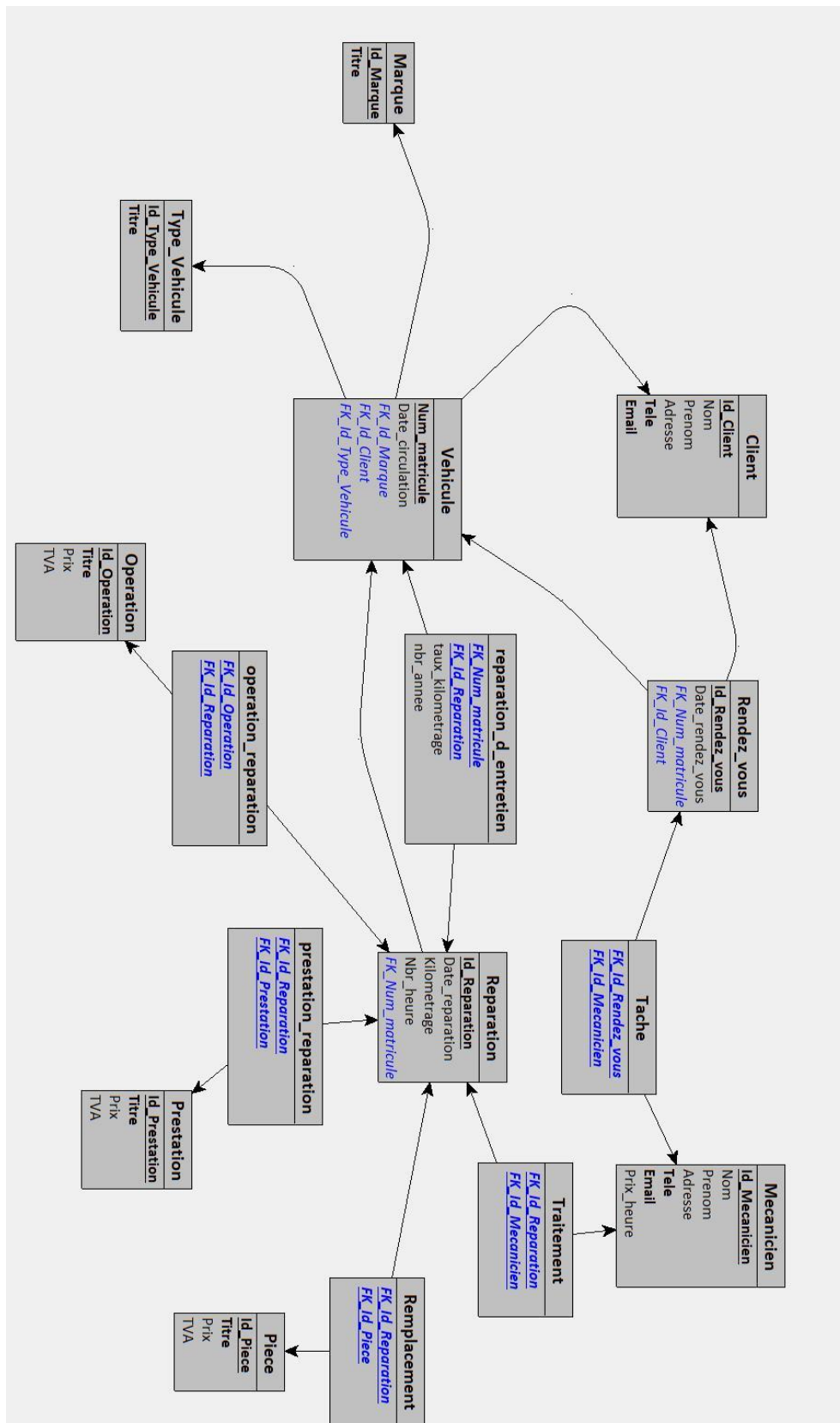
### 3. Le modèle conceptuel des données MCD

Le modèle MCD est un regroupement de toutes les entités et les associations entre eux. C'est une représentation statique, structurée du système d'information d'une entreprise.



#### 4. Le modèle logique des données MLD

Le modèle MLD est un regroupement de toutes les relations/tables de la base de données. C'est la modélisation logique des données qui tient compte de l'organisation des données.



## 5. Le modèle relationnelle (MLD textuelle) :

Client = (Id\_Client, Nom, Prenom, Adresse, **Tele**, Email);  
Type\_Vehicule = (Id\_Type\_Vehicule, Titre);  
Operation = (Id\_Operation, Titre, Prix, TVA);  
Piece = (Id\_Piece, Titre, Prix, TVA);  
Marque = (Id\_Marque, Titre);  
Mecanicien = (Id\_Mecanicien, Nom, Prenom, Adresse, **Tele**, Email, Prix\_heure);  
Prestation = (Id\_Prestation, Titre, Prix, TVA);  
Vehicule = (Num\_matricule, Date\_circulation, #FK\_Id\_Marque, #FK\_Id\_Client, #FK\_Id\_Type\_Vehicule);  
Rendez\_vous = (Id\_Rendez\_vous, Date\_rendez\_vous, #FK\_Num\_matricule, #FK\_Id\_Client);  
Reparation = (Id\_Reparation, Date\_reparation, Kilometrage, Nbr\_heure, #FK\_Num\_matricule);  
Traitement = (#FK\_Id\_Reparation, #FK\_Id\_Mecanicien);  
operation\_reparation = (#FK\_Id\_Operation, #FK\_Id\_Reparation);  
Remplacement = (#FK\_Id\_Reparation, #FK\_Id\_Piece);  
prestation\_reparation = (#FK\_Id\_Reparation, #FK\_Id\_Prestation);  
reparation\_d\_entretien = (#FK\_Num\_matricule, #FK\_Id\_Reparation, taux\_kilometrage, nbr\_annee);

## 6. Normalisation

Le but essentiel de la normalisation est d'éviter les anomalies transactionnelles pouvant découler d'une mauvaise modélisation des données et ainsi éviter un certain nombre de problèmes potentiels tels que les anomalies de lecture, les anomalies d'écriture, la redondance des données et la contre-performance. »  
-- Source : Wikipédia

### ❖ Vérifions que la base de données est en 3ème forme normale:

- ✓ **1NF:**  
Tous les attributs de toutes les relations sont des valeurs atomiques donc nous sommes en 1NF.
- ✓ **2NF:**  
Dans chacune de nos relations, les attributs non clés ne dépendent pas d'une partie de la clé donc nous sommes bien en 2NF.
- ✓ **3NF :**  
Dans toutes nos relations, aucun attribut non clé ne dépend pas d'autres attributs n'appartenant pas à la clé, donc nous sommes bien en 3NF.

De plus nous avons utilisé la **méthode Merise** pour développer notre base de données, nous sommes donc forcément en 3 NF.  
En effet, pour vérifier que la base de données est en 3ème forme normale il faut que chaque relation, de la MLD, soit en 3ème forme normale.

### ❖ Exemple de vérification:

✓ **Reparation** = (Id\_Reparation, Date\_reparation, kilometrage, nbr\_heure, #Fk\_Num\_matricule);

Les dépendances fonctionnelles :

F : {

Id\_Reparation → DateDepart

Id\_Reparation → Date\_reparation

Id\_Reparation → kilometrage

Id\_Reparation → nbr\_heure

Id\_Reparation → Fk\_Num\_matricule

Date\_reparation, kilometrage, nbr\_heure, #Fk\_Num\_matricule → IdReservation

}

On calcule la fermeture transitive d'Id\_Reparation :

(Id\_Reparation)<sup>+</sup> : Id\_Reparation, Date\_reparation, kilometrage, nbr\_heure, #Fk\_Num\_matricule

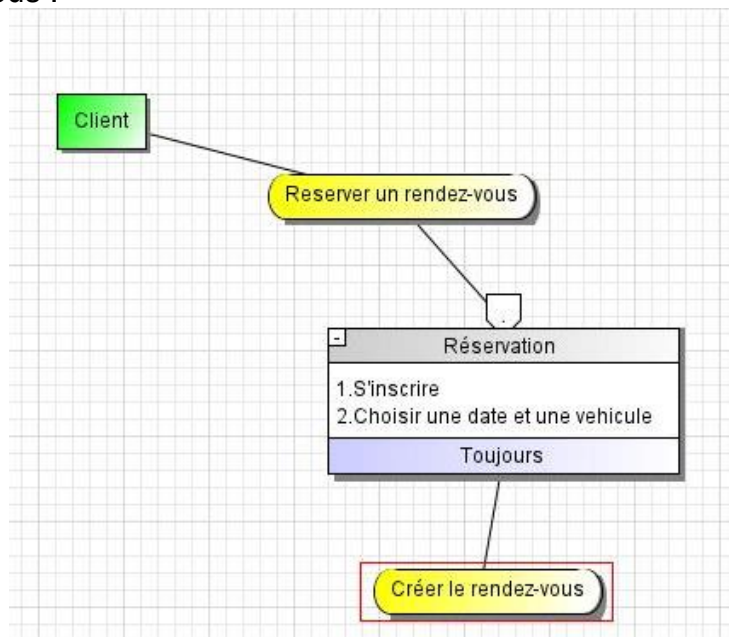
On retrouve bien tous les attributs de la relation donc IdReservation est une clé.

Toutes les dépendances fonctionnelles sont des valeurs atomiques donc nous sommes en 1NF. Aucun attribut non clés ne dépend pas d'une partie de la clé, donc nous sommes bien en 2NF. Aucun attribut non clé ne dépend pas d'autres attributs non clé, donc nous sommes bien en **3NF**.

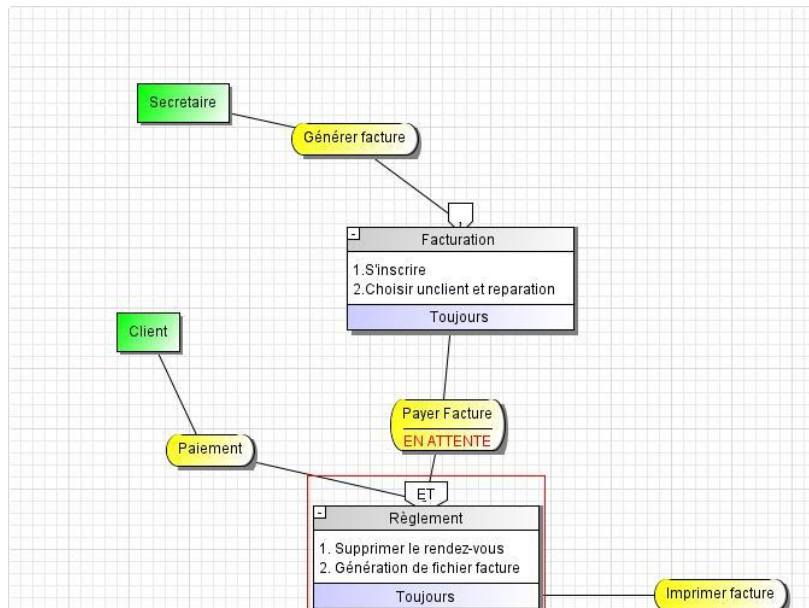
## 7. Modèle conceptuel de traitement (MCT)

**Modèle conceptuel de traitement** ou MCT est la représentation dynamique du système d'information. Il permet de savoir quoi faire selon l'évènement en cours.

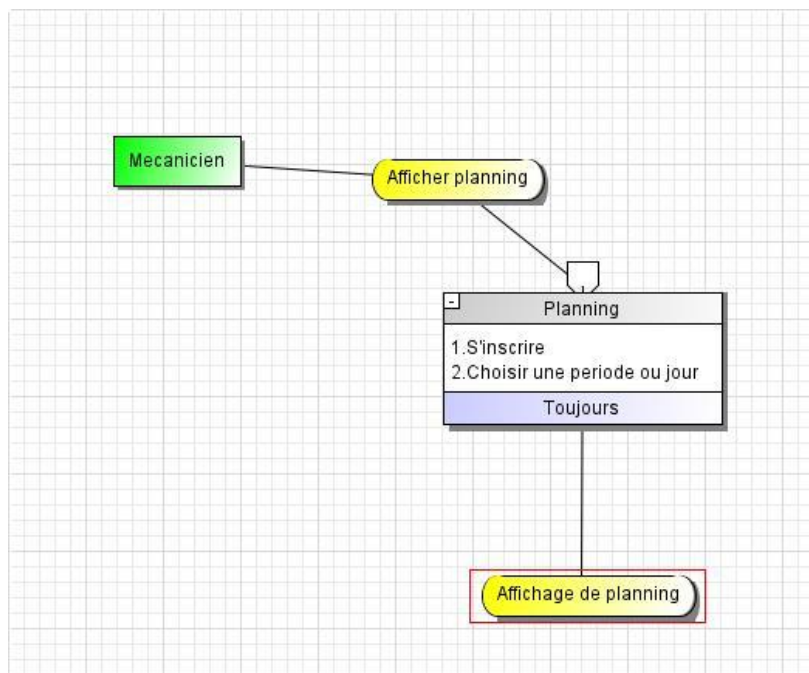
MCT Rendez-vous :



MCT facturation :

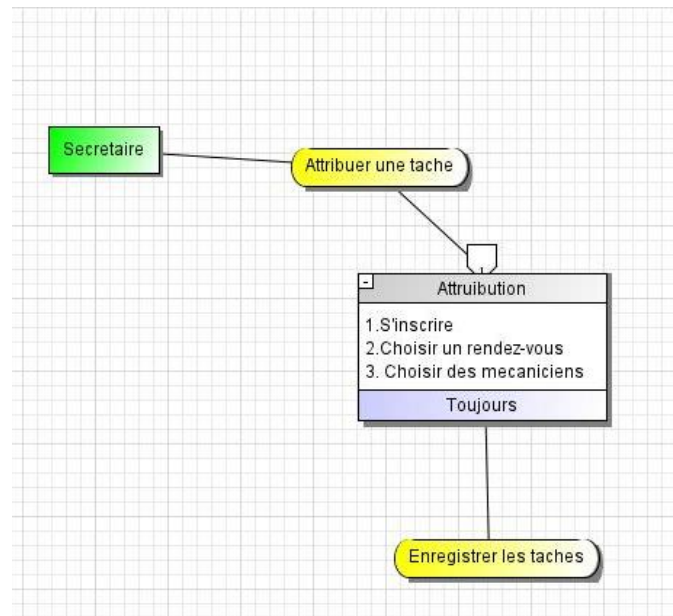


MCT planning mécanicien :





MCT attribution des taches :



### III. Partie SQL et Algèbre relationnelle

Dans cette partie, nous décrivons comment on a fait pour la création de la base de données et tester les fichiers SQL joints au rapport.

Ici on présente une liste non exhaustive des requêtes possibles. Reste à savoir que selon l'interface de l'application et ses besoins on peut en proposer d'autre.

#### R1 : La liste des clients avec leurs véhicules

**R1 : Donner la liste des clients avec leurs véhicules.**

SQL :

```
-- R1 : Donner la liste des clients avec leurs véhicules.
SELECT c.Nom || ' ' || c.Prenom AS nom_complet, c.Email,
       v.Num_matricule, m.Titre AS Marque_, t.titre type
FROM client c
JOIN vehicule v
ON v.FK_Id_Client = c.Id_Client
JOIN marque m
ON v.fk_id_marque = m.id_marque
JOIN type_vehicule t
ON v.FK_Id_Type_Vehicule = t.Id_Type_Vehicule;
```

### Algèbre Relationnelle :

Les clients et leurs véhicules:

$T1 : ((Client) \bowtie_{IdClient} Vehicule)$

Ajouter la marque et le type du véhicule :

$T2 : ((Marque \bowtie_{IdMarque} T1) \bowtie_{IdType} Type\_vehicule)$

Sélection des informations nécessaires:

$R1 : \pi_{\begin{matrix} NomClient \\ PrenomClient \\ email \\ Matricule \\ Marque \\ Type \end{matrix}}(T2)$

### R2 : La liste des rendez-vous clients de la journée

**R2 : Liste des rendez-vous clients de la journée.** On décide par une date de rendez-vous égale à d'aujourd'hui.

### SQL :

```
-- R2 |: La liste des rendez-vous clients de la journée.
WITH rendez_vous_aujourd'hui AS (
    SELECT r.*
    FROM rendez_vous r
    WHERE r.Date_rendez_vous = CURDATE()
)

SELECT r.Date_rendez_vous, c.Nom, c.Prenom, c.Email,
       v.Num_matricule, m.Titre AS Marque_, t.titre type
FROM rendez_vous_aujourd'hui r
JOIN vehicule v
ON r.FK_Num_matricule = v.Num_matricule
JOIN client c
ON v.FK_Id_Client = c.Id_Client
JOIN marque m
ON v.fk_id_marque = m.id_marque
JOIN type_vehicule t
ON v.FK_Id_Type_Vehicule = t.Id_Type_Vehicule;
```

### Algèbre Relationnelle :

Les rendez-vous d'aujourd'hui et leurs véhicules:

$T1 : (\sigma_{Date\_rendez\_vous = dateAujourd'hui}(Rendez\_vous) \bowtie_{Num\_matricule} Vehicule)$

Ajouter la marque et le type du véhicule :

$T2 : ((Marque \bowtie_{IdMarque} T1) \bowtie_{IdType} Type\_vehicule)$

Ajouter le client :

T3 :  $(T2 \bowtie_{\text{IdClient}} \text{Rendez\_vous})$

Sélection des informations nécessaires:

R2 :  $\pi_{\text{Date\_rendez\_vous}}(T3)$   
 $\pi_{\text{NomClient}}$   
 $\pi_{\text{PrenomClient}}$   
 $\pi_{\text{email}}$   
 $\pi_{\text{Matricule}}$   
 $\pi_{\text{Marque}}$   
 $\pi_{\text{Type}}$

### R3 : La liste des rendez-vous clients de la semaine

**R3 : Liste des rendez-vous clients de la semaine.** On décide par une date de rendez-vous entre d'aujourd'hui et la date après 7 jours.

#### SQL :

```
-- R3 : La liste des rendez-vous clients de la semaine.
WITH rendez_vous_aujourd'hui AS (
    SELECT *
    FROM rendez_vous
    WHERE Date_rendez_vous BETWEEN CURDATE() AND DATE_ADD(CURDATE(), INTERVAL 7 DAY)
)

SELECT r.Date_rendez_vous, c.Nom, c.Prenom, c.Email,
       v.Num_matricule, m.Titre AS Marque_, t.titre type
FROM rendez_vous_aujourd'hui r
JOIN vehicule v
ON r.FK_Num_matricule = v.Num_matricule
JOIN client c
ON v.FK_Id_Client = c.Id_Client
JOIN marque m
ON v.fk_id_marque = m.id_marque
JOIN type_vehicule t
ON v.FK_Id_Type_Vehicule = t.Id_Type_Vehicule;
```

#### Algèbre Relationnelle :

Les rendez-vous de la semaine et leurs véhicules:

T1 :

$(\sigma_{\text{Aujourd'hui} < \text{Date\_rendez\_vous} < \text{Aujourd'hui} + 7}(\text{Rendez\_vous}) \bowtie_{\text{Num\_matricule}} \text{Vehicule})$

Ajouter la marque et le type du véhicule :

T2 :  $((\text{Marque} \bowtie_{\text{IdMarque}} T1) \bowtie_{\text{IdType}} \text{Type\_vehicule})$

Ajouter le client :

T3 :  $(T2 \bowtie_{\text{IdClient}} \text{Rendez\_vous})$

Sélection des informations nécessaires:

**R2 :  $\pi_{Date\ rendez-vous}(T3)$**

*NomClient*  
*PrenomClient*  
*email*  
*Matricule*  
*Marque*  
*Type*

## R4 : Le planning des mécaniciens

**R4 : Donner le planning des mécaniciens.**

SQL :

```
-- R4: Le planning des mécaniciens
SELECT  r.Date_rendez_vous, m.Nom, m.Prenom, m.Email,
        v.Num_matricule, ma.Titre AS Marque_, ty.titre type
FROM    mecanicien m
JOIN    tache t
ON      m.Id_Mecanicien = t.FK_Id_Mecanicien
JOIN    rendez_vous r
ON      t.FK_Id_Rendez_vous = r.Id_Rendez_vous
JOIN    vehicule v
ON      v.Num_matricule = r.FK_Num_matricule
JOIN    marque ma
ON      v.fk_id_marque = ma.id_marque
JOIN    type_vehicule ty
ON      v.FK_Id_Type_Vehicule = ty.Id_Type_Vehicule;
```

## Algèbre Relationnelle :

Les mécaniciens et leurs tâches et rendez-vous attribués:

**T1 :  $(Mecanicien) \bowtie_{IdMecanicien} Tache \bowtie_{IdRendez\_vous} Rendez\_vous$**

Ajouter le véhicule :

**T2 :  $(T1 \bowtie_{IdNum\_Matricule} Vehicule)$**

Ajouter la marque et le type du véhicule :

**T3 :  $((Marque \bowtie_{IdMarque} T2) \bowtie_{IdType} Type\_vehicule)$**

Sélection des informations nécessaires:

**R4 :  $\pi_{Date\ rendez-vous}(T3)$**

*NomMecanicien*  
*PrenomMecanicien*  
*email*  
*Matricule*  
*Marque*  
*Type*

## R5 : La facturation d'une réparation

**R5 : Donner la facture d'une réparation d'un véhicule.** On choisit la réparation et on inclut les prix des opérations, prestations, pièces et la main d'œuvre.

### SQL :

```
-- R5: La facturation de la réparation
WITH reparation_en_question AS (
  SELECT *
  FROM reparation
  WHERE id_reparation = 2 -- id de la reparation en question
)

SELECT rep.id_reparation, rep.date_reparation, (rep.nbr_heure * me.Prix_heure) AS main_doeuvre,
       op.Titre operation_, op.Prix prx_operation, op.TVA tva_operation,
       pre.Titre prestation_, pre.Prix prx_prestation, pre.TVA tva_prestation,
       pie.Titre piece_, pie.Prix prx_piece, pie.TVA tva_piece
FROM reparation_en_question rep
JOIN traitement tr
ON tr.FK_Id_Reparation = rep.id_reparation
JOIN mecanicien me
ON me.id_mecanicien = tr.FK_Id_Mecanicien
JOIN remplacement re
ON re.FK_Id_Reparation = rep.id_reparation
JOIN piece pie
ON pie.Id_Piece = re.FK_Id_Piece
JOIN prestation_reparation pr_rep
ON pr_rep.FK_Id_Reparation = rep.id_reparation
JOIN prestation pre
ON pre.Id_Prestation = pr_rep.FK_Id_Prestation
JOIN operation_reparation op_rep
ON op_rep.FK_Id_Reparation = rep.id_reparation
JOIN operation op
ON op.Id_Operation = op_rep.FK_Id_Operation;
```

### Algèbre Relationnelle :

La réparation X = 2 avec les mécaniciens concernés:

**T1 :**

$\sigma_{Id\_reparation=2} (Reparation) \bowtie_{IdReparation} Traitement \bowtie_{IdMecanicien} Mecanicien$

Ajouter les pièces s'ils existent :

**T2 :**  $(T1 \bowtie_{IdReparation} Remplacement \bowtie_{IdPiece} Piece)$

Ajouter les prestations s'ils existent :

**T3 :**  $(T2 \bowtie_{IdReparation} Prestation\_reparation \bowtie_{IdPrestation} Prestation)$

Ajouter les opérations :

**T4 :**  $(T3 \bowtie_{IdReparation} Operation\_reparation \bowtie_{IdOperation} Operation)$

Sélection des informations nécessaires:

**R5 :  $\pi$**  **(T4)**

|  |
|--|
| <i>IdReparation</i>                            |
| <i>Date_reparation</i>                         |
| <i>(PrixHeure * nbr_heure) AS main_doeuvre</i> |
| <i>titreOperation</i>                          |
| <i>prixOperation</i>                           |
| <i>TVAOperation</i>                            |
| <i>titrePrestation</i>                         |
| <i>prixPrestation</i>                          |
| <i>TVAPrestation</i>                           |
| <i>titrePiece</i>                              |
| <i>prixPiece</i>                               |
| <i>TVAPiece</i>                                |

## R6 : La liste des réparations faites sur un véhicule depuis un temps donné

**R6 : Donner la liste des réparations faites sur un véhicule depuis un temps donné (ici, il y a 2 mois et 4 jours).** On récupère le minimum d'informations nécessaire pour la réparation. On peut toutefois avoir recours à la jointure pour récupérer les informations des pièces, opérations et prestations.

### SQL :

```
-- R6: La liste des réparations faites sur un véhicule depuis un temps donné
WITH reparation_vehicule_en_question AS (
    SELECT r.*
    FROM reparation r
    WHERE r.FK_Num_matricule = 'AA-001-AA' -- id de la vehicule en question
    AND r.Date_reparation >= DATE_SUB((DATE_SUB(CURDATE(), INTERVAL 2 MONTH)), INTERVAL 4 DAY)
)

SELECT r.Id_Reparation, v.Num_matricule, m.Titre AS Marque_, t.titre TYPE,
       r.Date_reparation, r.Kilometrage, r.Nbr_heure
FROM reparation_vehicule_en_question r
JOIN vehicule v
ON v.Num_matricule = r.FK_Num_matricule
JOIN marque m
ON v.fk_id_marque = m.id_marque
JOIN type_vehicule t
ON v.FK_Id_Type_Vehicule = t.Id_Type_Vehicule;
```

### Algèbre Relationnelle :

Les activités avec leurs animateurs:

T1 :  $\sigma_{Id\_reparation=1} \left( \sigma_{Date\_reparation > DateX} (Reparation) \right)$

Ajouter le véhicule :

T2 :  $(T1 \bowtie_{IdNum\_Matricule} Vehicule)$

Ajouter la marque et le type du véhicule :

T3 :  $\left( (Marque \bowtie_{IdMarque} T2) \bowtie_{IdType} Type\_vehicule \right)$

Sélection des informations nécessaires:

**R4 :  $\pi$**  *Id\_reparation* (**T3**)  
*NumMatricule*  
*Date\_reparation*  
*kilometrage*  
*nbr\_heure*  
*Marque*  
*Type*

## R7 : Le montant total de la facturation d'une réparation

**R7 : Donner le montant total de la facture d'une réparation d'un véhicule.** On choisit la réparation et on inclut les prix des opérations, prestations, pièces et la main d'œuvre.

### SQL :

```
-- R7: Le montant total de la facturation d'une réparation
WITH reparation_en_question AS (
    SELECT *
    FROM reparation
    WHERE id_reparation = 2 -- id de la reparation en question
)

SELECT rep.id_reparation, rep.date_reparation,
       ((rep.nbr_heure * me.Prix_heure) +
        op.Prix + pre.Prix + pie.Prix ) AS 'Sous Total HT',
       (op.TVA * op.Prix+pre.TVA* pre.Prix+pie.TVA * pie.Prix)/100 AS 'taux TVA Total',
       (rep.nbr_heure * me.Prix_heure) +
       (op.Prix + op.TVA/100 * op.Prix) +
       (pre.Prix + pre.TVA/100* pre.Prix)+
       (pie.Prix + pie.TVA/100 * pie.Prix) AS 'Montant Total TTC'
FROM reparation_en_question rep
JOIN traitement tr
ON tr.FK_Id_Reparation = rep.id_reparation
JOIN mecanicien me
ON me.id_mecanicien = tr.FK_Id_Mecanicien
JOIN remplacement re
ON re.FK_Id_Reparation = rep.id_reparation
JOIN piece pie
ON pie.Id_Piece = re.FK_Id_Piece
JOIN prestation_reparation pr_rep
ON pr_rep.FK_Id_Reparation = rep.id_reparation
JOIN prestation pre
ON pre.Id_Prestation = pr_rep.FK_Id_Prestation
JOIN operation_reparation op_rep
ON op_rep.FK_Id_Reparation = rep.id_reparation
JOIN operation op
ON op.Id_Operation = op_rep.FK_Id_Operation;
```

### Algèbre Relationnelle :

La réparation X = 2 avec les mécaniciens concernés:

**T1 :**

$\sigma_{Id\_reparation=2} (Reparation) \bowtie_{IdReparation} Traitement \bowtie_{IdMecanicien} Mecanicien$

Ajouter les pièces s'ils existent :

$T2 : (T1 \bowtie_{IdReparation} Remplacement \bowtie_{IdPiece} Piece)$

Ajouter les prestations s'ils existent :

$T3 : (T2 \bowtie_{\text{IdReparation}} \textit{Prestation\_reparation} \bowtie_{\text{IdPrestation}} \textit{Prestation})$

Ajouter les opérations :

$T4 : (T3 \bowtie_{\text{IdReparation}} \textit{Operation\_reparation} \bowtie_{\text{IdOperation}} \textit{Operation})$

Sélection des informations nécessaires:

**R5 :  $\pi$**  *IdReparation* (**T4**)  
*Date\_reparation*  
*TotalPrixHT*  
*TotalTVA*  
*TotalPrixTTC*

## R8 : Liste des réparations d'entretien d'un véhicule

**R8 : Donner les réparations d'entretien d'un véhicule.**

SQL :

```
-- R8: Liste des reparations d'entretien d'un vehicule
WITH reparations_en_question AS (
  SELECT r.*
  FROM reparation_d_entretien r
  WHERE r.FK_Num_matricule = 'AA-001-AA'
)

SELECT rep.id_reparation, rep.date_reparation, r.taux_kilometrage AS cycle_kelometrage, r.nbr_annee AS cycle_annee
FROM reparations_en_question r
JOIN reparation rep
ON rep.Id_Reparation = r.FK_Id_Reparation;
```

Algèbre Relationnelle :

Les réparations d'entretien de la véhicule X = AA – 001 – AA:

**T1 :  $\sigma$**  *Num\_Matricule= AA-001-AA* (**Reparation\_d\_entretien**)

Ajouter les réparations correspondantes :

$T2 : (T1 \bowtie_{\text{IdReparation}} \textit{Reparation})$

Sélection des informations nécessaires:

**R8 :  $\pi$**  *IdReparation* (**T2**)  
*Date\_reparation*  
*cycle\_kilometrage*  
*cycle\_annee*



## Conclusion

Nous avons donc notre base de données permettant la gestion d'un garage, que nous avons développé grâce à la méthode merise. Nous n'avons cependant pas traité la partie sécurité et quelques diagrammes.

Cependant il y a un point litigieux en ce qui concerne la sécurité. Le dossier client contient des informations confidentielles, et l'administrateur à la vision sur tout, or il ne devrait pas. Mais à ce jour il est impossible de bloquer la vue ou l'accès à l'administrateur car c'est lui qui régit toute l'organisation de la base de données, donc on supposera que l'administrateur sait garder les informations comme un secret.

Comme perspectives, nous envisageons de réaliser un site web qui interagit avec la base et de réaliser une API qui permettra la possibilité de travailler avec la même base avec une application mobile.

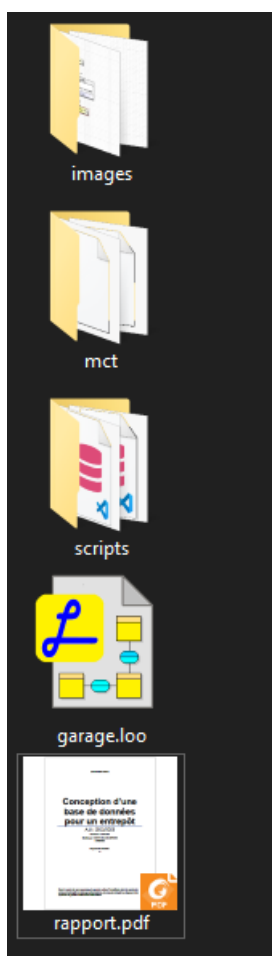
Donc on aurait eu les droits suivants :

|                       | <b>Droit lecture</b>                                       | <b>Droit écriture</b>                  |
|-----------------------|--|--|
| <b>Administration</b> | Toutes les tables  | Toutes les tables                      |
| <b>Mécanicien</b>     | Toutes les tables  | Son profile, ses taches et réparations |
| <b>Secrétaire</b>     | Toutes les tables  | Rendez-vous, facturation               |
| <b>Client</b>         | Toutes les tables concernant les informations personnelles | Les tables de leur profile.            |

# Annexe

Le rapport est joint par:

- Des fichiers des diagrammes (looping, JMCT) ;
- Scripts SQL :
  - ✓ Script création base et tables ;
  - ✓ Script insertion données ;
  - ✓ Script des requêtes de teste ;
  - ✓ Script pour la suppression des tables.



## Webographie

- [1] <https://www.sumup.com/fr-fr/factures/essentiels-facturation/modele-de-facture-garage-automobile/>
- [2] <https://www.younited-credit.com/projets/credit-auto/infos/types-reparations-voiture-cout>
- [3] <https://www.fiches-auto.fr/articles-auto/entretien-automobile/s-470-tarifs-main-d-oeuvre.php>
- [4] <https://www.tiregom.fr/blog/quels-types-de-prestations-proposent-un-garage-auto-1808.html>
- [5] [https://fr.wikipedia.org/wiki/Plaque\\_d%27immatriculation\\_fran%C3%A7aise](https://fr.wikipedia.org/wiki/Plaque_d%27immatriculation_fran%C3%A7aise)
- [6] <https://stackoverflow.com/questions/7568072/subtract-month-and-day-mysql>