

|  |  |
| --- | --- |
| ***Entreprise :*** | *GSB Laboratoire* |
| ***Réalisateur :*** | *Ilyas MOHETNA* |
| ***Date du rapport :*** | *20/09/2023* |

Table des matières

[1) Historique du document : 3](#_Toc149158019)

[2) Contexte du projet : 4](#_Toc149158020)

[3) Le domaine étudié : 4](#_Toc149158021)

[a) Domaine de l’entreprise : 4](#_Toc149158022)

[b) Domaine de développement : 4](#_Toc149158023)

[4) Le fonctionnement du système : 5](#_Toc149158024)

[a) Les modules : 5](#_Toc149158025)

[b) L’accès à l’intranet : 5](#_Toc149158026)

[c) Le périmètre du système : 6](#_Toc149158027)

[5) Extensions possibles : 7](#_Toc149158028)

[6) La modélisation conceptuelle : 7](#_Toc149158029)

[A) Modèle de dictionnaire : 8](#_Toc149158030)

[B) Le modèle entité association : 10](#_Toc149158031)

[a. La présentation du modèle entité association 10](#_Toc149158032)

[b) Les explications sur les choix modélisation et cardinalités : 11](#_Toc149158033)

[7) La modélisation logique 14](#_Toc149158034)

[A) Le modèle logique des données relationnel : 14](#_Toc149158035)

[a. Le modèle textuel : 15](#_Toc149158036)

[b. Le modèle schématisé : 16](#_Toc149158037)

[B) La vérification des types de données et l’ordre des colonnes des clés primaires composées 17](#_Toc149158038)

[C) La normalisation : 18](#_Toc149158039)

[8) Script de création de la base de données 18](#_Toc149158040)

[9) Modélisation UML 24](#_Toc149158041)

# Historique du document :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **f** | **Date de mise à jour** | **Auteur** | **Parties modifiées** | **Description** |
| **V1.0** | 20/09/2023 | Ilyas MOHETNA | Tout le document | La création du document |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

# Contexte du projet :

Née de la fusion de deux titans de l'industrie pharmaceutique, le laboratoire Galaxy Swiss Bourdin représente une force majeure dans le paysage pharmaceutique mondial. D'un côté, nous avons Galaxy, un géant américain reconnu pour son expertise dans le traitement des maladies virales, notamment le SIDA et les hépatites. De l'autre, Swiss Bourdin, un conglomérat européen résultant lui-même de la fusion de trois laboratoires, est renommé pour ses médicaments conventionnels.

En 2009, ces deux entités ont décidé de joindre leurs forces, donnant naissance à Galaxy Swiss Bourdin, un leader mondial du secteur pharmaceutique. Bien que l'entité européenne de Galaxy Swiss Bourdin ait choisi Paris comme siège administratif, la maison mère de cette multinationale se situe à Philadelphie, aux États- Unis.

# Le domaine étudié :

### Domaine de l’entreprise :

Le domaine d'étude se focalise sur le secteur dynamique et complexe de l'industrie pharmaceutique, en mettant un éclairage particulier sur le laboratoire Galaxy Swiss Bourdin (GSB). Originaire de la fusion entre Galaxy, un spécialiste reconnu dans le traitement des maladies virales, et Swiss Bourdin, un conglomérat travaillant sur des médicaments conventionnels, GSB représente une convergence unique de compétences et d'expertises.

Au-delà de sa genèse, l'étude s'intéresse aux interactions de GSB avec les praticiens, notamment à travers les visites médicales. Ces rencontres, essentielles pour la promotion des produits de GSB, sont souvent scrutées pour leur potentiel manque de transparence. L'analyse se penche également sur les procédures post-visites, comme la rédaction de rapports détaillés et la déclaration des frais associés par les employés.

Ainsi, ce domaine d'étude offre un aperçu approfondi des opérations, des défis et des opportunités auxquels GSB est confronté au sein du paysage pharmaceutique contemporain.

### Domaine de développement :

Dans le cadre du développement d’une solution technique pour l’ensemble de problème informatique de GSB , une application web un ensemble d’outils sera utiliser pour le développement :

* + - Langage de programmation et framework :
      * PHP 8.0
      * Laravel 10
      * HTML / CSS
      * Javascript
      * JQuery
      * cURL
    - Serveur et administrateur de base de donnée :
      * Maria DB 10.4.24
      * PHPMyAdmin 5.2.0
    - Outil de versionning :
      * Github 3.10.0
    - Serveur Web :
      * Développement :

Développement sera fait sous un ordinateur windows avec un serveur Apache le biais de XAMPP. Ainsi la mise en place de PHP et MariaDB.

* + - * Production :

Le déploiement de l’application web sera fait sous un serveur Linux dont un serveur web Apache avec les même caractéristique définis dans la cahier des charges (version des outils , langage installées …)

*La sélection des outils pour le développement de la solution pour GSB s'est basée sur la combinaison de robustesse, d'efficacité et de flexibilité. Le choix de PHP 8.0 et du Framework Laravel 10 repose sur leur réputation d'être puissants pour le développement d'applications web tout en offrant une grande facilité d'utilisation et une large communauté de soutien. HTML/CSS et Javascript, combinés à JQuery, garantissent une interface utilisateur réactive et interactive. L'utilisation de cURL facilite la communication avec d'autres services via HTTP, ce qui est essentiel pour une application intégrée. En ce qui concerne la gestion des données, MariaDB 10.4.24 est reconnue pour sa rapidité et sa fiabilité, et PHPMyAdmin 5.2.0 offre une interface intuitive pour l'administration de la base de données. Le choix de Github 3.10.0 pour la gestion de version assure une collaboration fluide entre les développeurs et un suivi clair des modifications. Enfin, l'utilisation de XAMPP pour le développement local permet une simulation précise de l'environnement de production, tandis que le déploiement sur un serveur Linux garantit stabilité et performance pour l'application en production.*

# Le fonctionnement du système :

## Les modules :

Pour résoudre l’ensemble de problèmes rencontré par le laboratoire GSB une solution web sera mise en place sous forme d’un intranet qui va regrouper l’ensemble d’application métier qui seront à destination des collaborateurs. Pour construire l’intranet 5 modules principales ont été mise en place pour de divers problèmes technique rencontré :

* + - La gestion des visites
    - La gestion des frais
    - La réservation des salles
    - Le covoiturage
    - Le paramétrage et l’administration

## L’accès à l’intranet :

L’accès à l’intranet sera à travers une combinaison **email** , **mot de passe** et destinée à chaque employeur dans l’entreprise notamment : Les visiteurs , Les comptables , Les délégués régionaux , Les responsables de secteurs.

## Le périmètre du système :

* + - La gestion des visites :
      * La déclaration d’une visite
      * La modification d’une visite
      * La clôture d’une visite
      * La création d’un rapport de la visite
      * La notification du délégué lors de la création et la fin d’une visite
      * La génération automatique d’un fichier PDF ordre de mission
      * Page d’ensemble pour le suivi / la recherche avec filtres des anciennes/nouvelles visites.
    - La gestion des frais :
* Partie visiteur :
  + La création d’une note de frais (Hôtel , Train, Dépenses hors forfait …) en la liant à une visite.
  + La suppression d’une note de frais
  + Page d’ensemble pour le suivi / la recherche avec filtres de toutes notes de frais.
* Partie comptable :
  + L’accès au tableau d’ensemble de notes de frais de tous les visiteurs
  + L’approbation , le refus d’une note de frais
  + L’accès à l’ensemble de détail d’une note de frais (montant , nature , quantité , date , justificatives)
    - La réservation des salles :
      * Recherche rapide de disponibilité d’une salle
      * Dans le cas si la salle est disponible une possibilité de réservation sera proposé
      * Durant la réservation ; une liste de choix (Repas demandé , Projecteur , Accès à internet) sera proposé autres que les informations de bases (Date de réservation , Motif de réservation , Nombre total de personnes présentes)
      * La confirmation par mail après la réservation
      * Page d’ensemble pour le statut de réservation de toutes les salles
    - Le covoiturage :
      * Le module sera accessible à tous collaborateur au seins du groupe GSB.
      * Possibilité de proposé des trajets avec le véhicule personnel de l’employer sous condition de la délivrance d’un permis de conduite valide et la carte grise et le pré ajout du véhicule soit avec le véhicule de service de l’entreprise.
      * La possibilité de définir le trajet et les arrêts desservis
      * La possibilité de supprimer une offre de covoiturage
      * La possibilité de recherche approfondi des offres de covoiturages des autres employés.
      * La possibilité d’accès au détail de réservation
      * L’accès au détail de l’offre et voir les personnes qui ont réservé le trajet.
      * Les notifications mail
      * L’intégration du module directement dans le cas de création d’une visite
    - Le paramétrage et l’administration
      * La gestion du personnel :
        + L’ajout d’un employé en désignant sa fonction et d’où ses droits d’accès.
        + La suppression d’un employé
      * La gestion de frais :
        + La configuration des montants de forfaits
      * La réservation de salles :
        + L’ajout de bâtiment sous une région
        + L’ajout de salles sous un bâtiment
      * Le covoiturage :
        + L’ajout de véhicule de service
      * Configuration générale :
        + L’ajout de famille de médicaments
        + L’ajout de médicament sous une famille

# Extensions possibles :

* Création d’un model statistiques (Visites par mois, Suivi de dépenses, Ratio des familles de médicaments …)
* Annulation et modification d’une note de frais sous conditions (elle ne soit pas vu par un comptable)
* IA : Détection automatique de détail d’une note de frais à travers un ticket justificative. Par exemple déposer un ticket de péage va engendrer la création automatiquement d’une note de frais avec le montant et la date inclus sur le ticket.

# La modélisation conceptuelle :

## Modèle de dictionnaire :

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Nom | Type | Nature | Longueur | Identifiant | Exemple de valeur |
| Employe\_\_employe | | | | | |
| code\_employe | E | N | 20 | OUI | 1 |
| prenom | E | A | 30 | - | GAUTIER |
| nom | E | A | 30 | - | DUPONT |
| utilisateur | E | A | 50 | - | GAUTIER.DUVALE |
| mot\_de\_passe | E | AN | 255 | - | $2y$10$ny2uulBo6crtt6it1BhQzuiP07lBswYfAOkbKOta8pOkaj/4gY3WC |
| email | E | AN | 100 | - | [Gautier.duvale@domaine.com](mailto:Gautier.duvale@domaine.com) |
| date\_naissance | E | DATE | 10 | - | 10/06/1970 |
| date\_embauche | E | DATE | 10 | - | 10/06/1995 |
| dernière\_connexion | E | DATETIME | 19 | - | 2023-07-15 09:23:45 |
| Employe\_\_fonction | | | | | |
| code\_fonction | E | N | 20 | OUI | 1 |
| nom\_fonction | E | AN | 50 | - | Responsable régional |
| Parametrage\_\_ville | | | | | |
| id\_ville | E | N | 20 | OUI | 1 |
| nom | E | A | 50 | - | Belfort |
| code\_postal | E | N | 5 | - | 90000 |
| ville\_longitude | E | N | 9 | - | 6.863849 |
| ville\_latitude | E | N | 8 | - | 47.639674 |
| Parametrage\_\_departement | | | | | |
| id\_departement | E | N | 20 | OUI | 90 |
| nom\_departement | E | A | 50 | - | TERRITOIRE DE BELFORT |
| Parametrage\_\_region | | | | | |
| id\_region | E | N | 10 | OUI | 1 |
| nom\_region | E | A | 50 | - | Bourgogne-Franche-Comté |
| Covoiturage\_\_etape | | | | | |
| id\_etape | E | N | 20 | OUI | 1 |
| ordre\_etape | E | N | 11 | - | 1 |
| date\_passage | E | DATETIME | 19 | - | 2023-07-15 09:23:45 |
| Covoiturage\_\_reservation | | | | | |
| id\_reservation | E | N | 20 | OUI | 1 |
| date\_reservation | E | DATETIME | 19 | - | 2023-07-15 09:23:45 |
| Covoiturage\_\_trajet | | | | | |
| id\_trajet | E | N | 20 | OUI | 1 |
| nombre\_place\_maximum | E | N | 11 | - | 5 |
| Covoiturage\_\_vehicule | | | | | |
| id\_vehicule | E | N | 20 | OUI | 1 |
| immatriculation | E | AN | 15 | - | GT-934-BA |
| marque | E | AN | 15 | - | VOLKSWAGEN |
| model | E | AN | 15 | - | SCIROCCO |
| annee\_model | E | N | 4 | - | 2014 |
| type\_vehicule | E | A | 7 | - | ENUM deux valeurs possible ‘perso’ et ‘service’ |
| Frais\_\_frais | | | | | |
| id\_frais | E | N | 20 | OUI | 1 |
| montant\_total | Ca | N | 16 | - | 534,23 |
| date\_frais | E | DATE | 10 | - | 2023-07-15 |
| type\_forfait | E | A | 10 | - | ENUM deux valeurs possible ‘forfait’ et ‘horsforfait’ |
| horsforfait\_libelle | E | AN | 50 | - | Ticket d’avion N212 |
| forfait\_quantite | Ca | N | 11 | - | 5 |
| commentaire | E | AN | 255 | - | La somme présente est le pour une double nuitée d’hôtel |
| appartenance\_mois | E | N | 2 | - | 06 |
| appartenance\_annee | E | N | 4 | - | 2023 |
| Frais\_\_nature | | | | | |
| id\_nature | E | N | 20 | OUI | 1 |
| intitule\_frais | E | AN | 50 | - | Forfait Repas + Nuit Hotel |
| montant\_forfait | Ca | N | 16 | - | 70.00 |
| Frais\_\_justificative | | | | | |
| id\_justif | E | N | 20 | OUI | 1 |
| justif\_chemin | E | AN | 255 | - | /storage/document/83JF438DEZKLSDFK234KL3K.pdf |
| justif\_extension | E | AN | 5 | - | pdf |
| justif\_mime | E | AN | 20 | - | application/pdf |
| Frais\_\_situation\_validation | | | | | |
| code\_situation | E | N | 20 | OUI | 1 |
| libelle\_situation | E | A | 50 | - | EN COURS DE TRAITEMENT |
| Salle\_\_agence | | | | | |
| id\_agence | E | N | 20 | OUI | 1 |
| nom\_agence | E | AN | 50 | - | AGENCE LAB123 PHARMATECH |
| Salle\_\_batiment | | | | | |
| id\_batiment | E | N | 20 | OUI | 1 |
| nom\_batiment | E | AN | 50 | - | BATIMENT PRIMAIRE 2 |
| Salle\_\_salle | | | | | |
| id\_salle | E | N | 20 | OUI | 1 |
| nom\_salle | E | AN | 20 | - | B204 |
| Salle\_\_extra | | | | | |
| id\_extra | E | N | 20 | OUI | 1 |
| libelle\_extra | E | AN | 100 | - | Jus d’orange |
| Salle\_\_materiel\_type | | | | | |
| id\_materiel | N | N | 20 | OUI | 2 |
| nom\_materiel | E | AN | 50 | - | CABLES RJ45 , PROJECTEUR … |
| Salle\_\_posession\_materiel | | | | | |
| id\_salle | E | N | 20 | OUI | 1 |
| id\_materiel | E | AN | 20 | OUI | 2 |
| Salle\_\_reservation | | | | | |
| id\_reservation | E | N | 20 | OUI | 1 |
| date\_debut\_reservation | E | DATETIME | 19 | - | 2023-07-15 09:23:45 |
| date\_fin\_reservation | E | DATETIME | 19 | - | 2023-07-15 15:23:45 |
| nombre\_de\_personnes | E | N | 16 | - | 5 |
| Visite\_\_visite | | | | | |
| id\_visite | E | N | 20 | OUI | 1 |
| date\_debut\_visite | E | DATETIME | 19 | - | 2023-07-15 09:23:45 |
| date\_fin\_visite | E | DATETIME | 18 | - | 2023-07-15 15:23:45 |
| rapport | E | AN | 255 | - | Un avis plutôt favorable émis par le praticien concernant … |
| Visite\_\_praticien | | | | | |
| id\_praticien | E | N | 20 | OUI | 1 |
| raison\_social | E | AN | 50 | - | MICHEL BERNANRD, CABINET BERNARD |
| prenom | E | A | 30 | - | MICHEL |
| nom | E | A | 30 | - | BERNARD |
| adresse | Co | AN | 100 | - | 26 RUE LA RENAISSANCE |
| Visite\_\_medicament | | | | | |
| id\_medicament | E | N | 20 | OUI | 1 |
| nom\_medicament | E | AN | 30 | - | Acétaminophène |
| photo\_medicament | E | AN | 255 | - | https://chemin\_vers\_photo\_Acétaminophène.com |
| Visite\_\_presentation\_medicament | | | | | |
| id\_visite | E | N | 20 | OUI | 1 |
| id\_medicament | E | N | 20 | OUI | 1 |
| Visite\_\_famille\_medicament | | | | | |
| id\_famille | E | N | 20 | OUI | 2 |
| nom\_famille | E | AN | 30 | - | Analgésiques |

## Le modèle entité association :

## La présentation du modèle entité association

Une image contenant texte, diagramme, carte, Plan

Description générée automatiquement

## b) Les explications sur les choix modélisation et cardinalités :

* + - **Employé (employé, fonction) :**

Plutôt que de créer des tables séparées pour différents types d'employés (par exemple, "Visiteur" et "Comptable"), nous avons opté pour une seule entité "Employé". Cette approche permet de stocker l'ensemble des données sur tous les employés qui auront accès à l'intranet, évitant ainsi les redondances inutiles de données et garantissant l'intégrité des informations.

*Les cardinalités :*

* + - * Un employé occupe une seule fonction
      * Un employé travaille dans une seule agence
      * Un employé peut avoir aucune ou plusieurs véhicules personnels
      * Un employé peut annoncer aucune ou plusieurs trajets de covoiturage
      * Un employé peut effectuer aucune ou plusieurs réservations de trajet
      * Un employé (visiteur) peut déclarer aucun ou plusieurs frais de mission
      * Un employé (comptable) peut valider aucune ou plusieurs notes de frais
      * Un employé (visiteur) peut faire aucune ou plusieurs visites
      * Un employé peut réserver aucune ou plusieurs salles dans tous les agences des différents départements ou régions.
    - **Paramétrage (Ville, Département, Région) :**

Le paramétrage pour la gestion des zones géographiques a été réalisé en tenant compte de la réalité, dans le but d'offrir un contrôle et une granularité optimaux. Notre choix s'est porté sur l'utilisation de villes en tant que clés étrangères, au lieu d'un champ texte libre, afin d'éviter les redondances de données, d'assurer la précision et d'éliminer les erreurs typographiques. Cette approche garantit également, à des fins statistiques, une présentation sans doublons des informations géographiques.

Pour alimenter les données de ces trois entités, nous prévoyons de tirer parti de l'API fournie par l'État SIREN. Cette source de données nous permettra d'obtenir un ensemble complet d'informations, y compris la longitude et la latitude des villes. Ces coordonnées géographiques sont essentielles pour le développement de fonctionnalités de cartographie, offrant la possibilité de géolocaliser avec précision les villes sur une carte.

Ce choix de modélisation garantit une base de données géographique fiable et précise, tout en facilitant l'intégration de données externes provenant de sources officielles, ce qui contribue à la qualité et à la cohérence des informations géographiques dans notre système.

*Les cardinalités :*

* + - * Une ville est rattachée à un département
      * Un département est situé dans une région
      * Une région contient plusieurs départements
      * Un département contient plusieurs villes
    - **Salle (Salle, Bâtiment, Agence, Réservation, Possession Matériel, Type Matériel, Extra)**

Le choix des entités "Salle", "Bâtiment" et "Agence" a été délibéré pour refléter au mieux la structure d'une agence et pour simplifier la création future de salles sous de nouveaux bâtiments ou agences. Cette modélisation permet de suivre de manière organique la hiérarchie des locaux au sein de l'entreprise, en garantissant une cohérence dans la gestion des salles.

L'entité "Matériel" a été introduite pour éviter les redondances de données. Elle permet de répertorier l'ensemble du matériel disponible, tout en offrant la possibilité de l'attribuer à différentes salles. Cela est réalisé grâce à l'association "Possède" (Possession Matériel), qui assure que chaque salle peut disposer de plusieurs équipements, ce qui optimise la gestion des ressources.

Pour gérer les réservations de salles, l'association "Réserver" a été créée. Elle joue un rôle essentiel en stockant les valeurs qui identifient la date de début de la réservation, la date de fin, ainsi que le nombre de personnes prévues. Cette association est également porteuse d'informations essentielles pour le traitement des extras. Par exemple, dans l'entité "Extra," des éléments tels que "Café" ou "Jus d'orange" peuvent être définis en tant que formules générales. La réservation permet de faire des choix spécifiques parmi ces options pour s'assurer que la quantité nécessaire est disponible, par exemple, en fonction du nombre de participants prévus.

Cette approche de modélisation permet de gérer efficacement les ressources liées aux locaux et aux équipements, tout en offrant une flexibilité pour créer, réserver et gérer des salles de manière intuitive au sein de l'entreprise.

*Les cardinalités :*

* + - * Une salle est abritée dans un seul bâtiment
      * Une agence contient un ou plusieurs bâtiments
      * Un bâtiment appartient à une seule agence
      * Un bâtiment contient une ou plusieurs salles
      * Une salle ne possède aucun ou plusieurs matériels
      * Un matériel peut être posséder par aucune ou par plusieurs salles
      * Une salle peut être réserver plusieurs fois (par un employé) « Sous conditions de différenciation de plage horaires »
      * Les extras peuvent êtres réserver dans aucune ou dans plusieurs réservations.
    - **Visite (Visite, Praticien, Médicament, Famille Médicament, Médicament présenté) :**

Dans le module "Visite", plusieurs choix de modélisation ont été délibérément effectués pour optimiser le développement et la gestion des données, ainsi que pour permettre une utilisation efficace des informations dans d'autres parties du système.

Entité "Praticien" Séparée : Le choix a été fait de créer une entité distincte pour les "Praticiens". Cette séparation vise à simplifier la gestion des informations sur les praticiens, en les distinguant clairement des employés. Cela permet également de faciliter le traitement des données liées aux visites médicales.

Entité "Visite" Centrale : Plutôt que d'utiliser une simple association entre les employés et les praticiens, une entité "Visite" a été créée. Cette approche centralise les informations relatives aux visites, offrant une plus grande flexibilité pour gérer les aspects spécifiques de chaque visite et simplifiant le suivi des données dans d'autres parties du système.

Entités "Médicament" et "Famille Médicament" pour une Meilleure Visibilité : Les entités distinctes "Médicament" et "Famille Médicament" ont été introduites pour offrir une visibilité plus claire lors de l'utilisation des données. Cela permet également de faciliter des analyses statistiques plus précises. Par exemple, il est possible d'étudier chaque branche de médicament de manière distincte pour obtenir des informations spécifiques.

Association "Médicament Présenté" : L'association "Médicament Présenté" a été créée pour permettre à un visiteur de présenter plusieurs médicaments lors de la même visite. Cette association facilite la gestion des informations sur les médicaments présentés lors des visites médicales et offre une souplesse pour les enregistrements complexes.

*Les cardinalités :*

* + - * Une visite peut être faite par un employé (visiteur)
      * Une visite n’engendre aucun ou plusieurs frais de mission
      * Lors d’une visite on peut présenter un ou plusieurs médicaments
      * Une visite concerne qu’un seul praticien
      * Un médicament est présenté dans plusieurs visites
      * Un médicament appartient à une seule famille de médicament
      * Une famille de médicament ne comporte aucun ou comporte plusieurs médicaments
      * Un praticien peut être visiter durant plusieurs visite (sous condition de plage horaire différentes)
      * Un praticien se localise dans une ville
    - **Frais (Frais, Justificative, Nature frais, Situation Validation) :**

Le choix de passer par une table "Frais" commune a été fait pour éviter les doublons et centraliser les informations du même module. Cette approche gère à la fois les "Frais Forfait" et les "Frais Hors Forfait" en distinguant le champ "type\_forfait" : les "Frais Forfait" auront des champs remplis tandis que les "Frais Hors Forfait" auront des champs nuls, et vice versa.

Pour gérer la validation des frais par le comptable et les retards, deux champs, "frais\_appartenance\_mois" et "frais\_appartenance\_annee", ont été ajoutés. Ils servent de pivot pour indiquer la période à laquelle les frais sont attribués. Ces champs reflètent initialement les mêmes valeurs que la date des frais, mais si un retard survient dans la fourniture de justificatifs, le mois peut changer pour reporter le frais au mois suivant.

Concernant les justificatifs, une entité distincte a été créée pour permettre à un frais d'avoir plusieurs justificatifs. Une table associée stocke l'ensemble des fichiers et est liée à travers l'identifiant du frais.

L'entité "Nature Frais" contient des forfaits prédéfinis avec des montants fixés par l'administration. Chaque forfait a son montant fixe, qui est multiplié par le champ "frais.forfait\_quantite" pour calculer le montant total dans la même table.

Enfin, l'entité "Situation Validation" contient les différents statuts de validation attribués aux frais. Ces statuts évoluent tout au long du processus de traitement, par exemple, "Créé", "En cours de traitement", "Validé", "En cours de remboursement", "Annulé", etc. Cette structure permet au comptable et au visiteur d'avoir des pages de suivi pour leurs frais, offrant une gestion efficace des processus de validation et de remboursement.

*Les cardinalités :*

* + - * Un frais est lié à une visite obligatoirement sans visite on ne peut pas déclarer de frais
      * Un frais est créé par un seul employé (visiteur)
      * Un employé (visiteur) peut créer un ou plusieurs frais
      * Un frais est validé par un employé (comptable)
      * Un justificative appartient à un seul frais
      * Un nature frais peut être la catégorie d’aucun ou de plusieurs frais
      * Une situation de validation peut être attribuer à plusieurs frais
      * Un frais peut avoir aucun ou plusieurs justificative (Forfait : Non obligatoire, Hors forfait : Obligatoire)
      * Un frais peut avoir 0 ou 1 nature frais (0 si le cas d’un frais hors forfait)
      * Un frais a une seule situation de validation
    - **Covoiturage (Trajet, Véhicule, Etape Covoiturage, Réservation)**

Annonces de Trajets et Étapes de Covoiturage : Le choix a été fait de diviser le processus de covoiturage en annonces de trajets et étapes de covoiturage distinctes. Les employés peuvent réserver des trajets en sélectionnant les étapes qui leur conviennent. Cette approche permet une flexibilité maximale pour les utilisateurs du système.

Entité "Véhicule" Complète : L'entité "Véhicule" a été créée pour regrouper tous les types de véhicules, à la fois ceux utilisés à des fins professionnelles et personnels. La distinction est basée sur le champ "type\_vehicule". Les employés peuvent effectuer une ou plusieurs réservations de véhicules dans différentes plages horaires, tout en garantissant que l'exclusion est respectée, ce qui signifie que la personne qui dépose l'annonce ne peut pas réserver son propre trajet.

Ces choix de modélisation visent à rendre le module de covoiturage intuitif et flexible pour les utilisateurs, en facilitant la recherche, la réservation de trajets et la gestion des véhicules, tout en respectant les contraintes d'exclusion pour une utilisation équitable du système.

*Les cardinalités :*

* + - * Un employé peut déposer 0 ou plusieurs trajets de covoiturage
      * Un employé peut réserver 0 ou plusieurs trajets de covoiturage sous deux conditions (des plages horaires différentes ainsi qu’il ne soit pas la même personne qui a déposé le trajet)
      * Un trajet utilise un seul véhicule obligatoirement
      * Un véhicule peut être ou pas utiliser dans un trajet
      * Un véhicule peut être en stock dans une agence (Le cas de véhicule de service)
      * Un véhicule peut appartenir à un employé (Le cas de véhicule personnel)
      * Un trajet contient la cardinalité particulière (2, N) pour bien préciser que le minimum d’un trajet est 2 étapes parce que le choix a été fait d’enregistrer les villes de départ et arrivé comme étape avec un ordre par exemple Belfort départ et Besançon arrivé on aura toujours au moins deux étapes covoiturage dans ce cas Belfort avec un ordre 1 et Besançon avec un ordre 2.
      * Une agence peut avoir aucune ou plusieurs voitures de service
      * Une étape de covoiturage est traverse obligatoirement une seule ville sur la table ville

# La modélisation logique

## Le modèle logique des données relationnel :



Dans le cadre de la conception de notre base de données, On a décidé d'appliquer des normes de normalisation dès l'étape du modèle entité-association. Cette décision découle de ma volonté de garantir la cohérence, l'efficacité et la qualité de notre base de données, tout en facilitant le processus de développement ultérieur. La normalisation des données est une pratique éprouvée qui vise à éliminer les anomalies potentielles, à réduire la redondance des données et à rendre la base de données plus flexible. En conséquence, j'ai veillé à ce que chaque entité et ses attributs soient correctement structurés, en éliminant les données répétées et en établissant des relations claires entre les entités. Cette démarche vise à garantir que notre base de données répondra aux besoins de l'application de manière optimale, tant en termes de performance que d'intégrité des données. Plus de détails sur le processus de normalisation seront abordés dans les paragraphes suivants, permettant ainsi de mieux comprendre comment nous avons optimisé notre modèle logique de données pour atteindre ces objectifs.[Citez votre source ici.]



## Le modèle textuel

* **parametrage\_\_region = (region\_id**, nom\_region**);**
* **employe\_\_fonction = (code\_fonction**, nom\_fonction**);**
* **frais\_\_nature = (id\_nature**, intitule\_frais, montant\_forfait**);**
* **visite\_\_famille\_medicament = (id\_famille**, nom\_famille**);**
* **frais\_\_situation\_validation = (code\_situation**, libelle\_situation**);**
* **salle\_\_extra = (id\_extra**, libelle\_extra**);**
* **salle\_\_materiel\_type = (id\_materiel**, nom\_materiel**);**
* **parametrage\_\_departement = (departement\_id**, nom\_departement *, #region\_id***);**
* **visite\_\_medicament = (id\_medicament**, nom\_medicament, photo\_medicament*, #id\_famille***);**
* **parametrage\_\_ville = (id\_ville**, nom, code\_postal, ville\_longitude, ville\_latitude*, #departement\_id***);**
* **visite\_\_praticien = (id\_praticien**, raison\_sociale, prenom, nom, adresse*, #id\_ville***);**
* **salle\_\_agence = (id\_agence**, nom\_agence *, #id\_ville***);**
* **salle\_\_batiment = (*#id\_agence*, id\_batiment**, nom\_batiment**);**
* **employe\_\_employee = (code\_emlpoye**, prenom, nom, utilisateur, mot\_de\_passe, email, date\_naissance, date\_embauche, date\_derniere\_connexion *, #id\_agence, #code\_fonction***);**
* **salle\_\_salle = (*#(#id\_agence, id\_batiment)*, id\_salle**, numero\_salle, superficie, capacite**);**
* **visite\_\_visite = (id\_visite**, date\_debut\_visite, date\_fin\_visite, rapport *, #id\_praticien, #code\_emlpoye***);**
* **frais\_\_frais = (id\_frais**, montant\_total, date\_frais, type\_forfait, horsforfait\_libelle, forfait\_quantite, commentaire, frais\_appartenance\_mois*,* frais\_appartenance\_annee*, #code\_emlpoye, #code\_situation, #code\_emlpoye\_1***\****, #id\_nature***\****, #id\_visite***);**
* **covoiturage\_\_vehicule = (id\_vehicule**, immatriculation, marque, model, annee\_model, type\_vehicule *, #id\_agence***\****, #code\_emlpoye***\*);**
* **frais\_\_justificative = (id\_justif**, justif\_chemin, justif\_extension, justif\_mime *, #id\_frais***);**
* **covoiturage\_\_trajet = (id\_trajet**, nombre\_place\_maximum *, #id\_vehicule, #code\_emlpoye***);**
* **covoiturage\_\_etape = (id\_etape**, ordre\_etape, date\_passage *, #id\_ville, #id\_trajet***);**
* **salle\_\_reservation = (*#code\_emlpoye, #(#(#id\_agence, id\_batiment), id\_salle), #id\_extra***, date\_debut\_reservation, date\_fin\_reservation, nombre\_de\_personnes**);**
* **visite\_\_presentation\_medicament = (*#id\_medicament, #id\_visite*);**
* **covoiturage\_\_reservation = (*#code\_emlpoye, #id\_trajet, #id\_etape***, date\_de\_reservation**);**
* **salle\_\_posession\_materiel = (*#(#(#id\_agence, id\_batiment), id\_salle), #id\_materiel*);**

NB : A REMPLIRE APRES

## Une image contenant texte, diagramme, capture d’écran, Plan Description générée automatiquementLe modèle schématisé

## 

## La vérification des types de données et l’ordre des colonnes des clés primaires composées

Une image contenant capture d’écran, carré, Rectangle, diagramme

Description générée automatiquement

## 

## La normalisation :

# Script de création de la base de données

DROP TABLE IF EXISTS `covoiturage\_\_etape`;

CREATE TABLE `covoiturage\_\_etape` (

  `id\_etape` bigint(20) unsigned NOT NULL AUTO\_INCREMENT,

  `ordre\_etape` int(11) NOT NULL,

  `date\_passage` datetime NOT NULL,

  `id\_trajet` bigint(20) unsigned NOT NULL,

  `id\_ville` bigint(20) unsigned NOT NULL,

  `created\_at` timestamp NULL DEFAULT NULL,

  `updated\_at` timestamp NULL DEFAULT NULL,

  PRIMARY KEY (`id\_etape`),

  KEY `covoiturage\_\_etape\_id\_etape\_index` (`id\_etape`),

  KEY `covoiturage\_\_etape\_ordre\_etape\_index` (`ordre\_etape`),

  KEY `covoiturage\_\_etape\_id\_trajet\_index` (`id\_trajet`),

  KEY `covoiturage\_\_etape\_id\_ville\_index` (`id\_ville`),

  CONSTRAINT `covoiturage\_\_etape\_id\_trajet\_foreign` FOREIGN KEY (`id\_trajet`) REFERENCES `covoiturage\_\_trajet` (`id\_trajet`),

  CONSTRAINT `covoiturage\_\_etape\_id\_ville\_foreign` FOREIGN KEY (`id\_ville`) REFERENCES `parametrage\_\_ville` (`id\_ville`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci;

LOCK TABLES `covoiturage\_\_etape` WRITE;

UNLOCK TABLES;

DROP TABLE IF EXISTS `covoiturage\_\_reservation`;

CREATE TABLE `covoiturage\_\_reservation` (

  `id\_reservation` bigint(20) unsigned NOT NULL AUTO\_INCREMENT,

  `date\_de\_reservation` datetime NOT NULL,

  `code\_employe` bigint(20) unsigned NOT NULL,

  `id\_trajet` bigint(20) unsigned NOT NULL,

  `id\_etape` bigint(20) unsigned NOT NULL,

  `created\_at` timestamp NULL DEFAULT NULL,

  `updated\_at` timestamp NULL DEFAULT NULL,

  PRIMARY KEY (`id\_reservation`),

  KEY `covoiturage\_\_reservation\_id\_reservation\_index` (`id\_reservation`),

  KEY `covoiturage\_\_reservation\_code\_employe\_index` (`code\_employe`),

  KEY `covoiturage\_\_reservation\_id\_trajet\_index` (`id\_trajet`),

  KEY `covoiturage\_\_reservation\_id\_etape\_index` (`id\_etape`),

  CONSTRAINT `covoiturage\_\_reservation\_code\_employe\_foreign` FOREIGN KEY (`code\_employe`) REFERENCES `employe\_\_employe` (`code\_employe`),

  CONSTRAINT `covoiturage\_\_reservation\_id\_etape\_foreign` FOREIGN KEY (`id\_etape`) REFERENCES `covoiturage\_\_etape` (`id\_etape`),

  CONSTRAINT `covoiturage\_\_reservation\_id\_trajet\_foreign` FOREIGN KEY (`id\_trajet`) REFERENCES `covoiturage\_\_trajet` (`id\_trajet`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci;

LOCK TABLES `covoiturage\_\_reservation` WRITE;

UNLOCK TABLES;

DROP TABLE IF EXISTS `covoiturage\_\_trajet`;

CREATE TABLE `covoiturage\_\_trajet` (

  `id\_trajet` bigint(20) unsigned NOT NULL AUTO\_INCREMENT,

  `nombre\_place\_maximum` int(11) NOT NULL,

  `id\_vehicule` bigint(20) unsigned NOT NULL,

  `code\_employe` bigint(20) unsigned NOT NULL,

  `created\_at` timestamp NULL DEFAULT NULL,

  `updated\_at` timestamp NULL DEFAULT NULL,

  PRIMARY KEY (`id\_trajet`),

  KEY `covoiturage\_\_trajet\_id\_trajet\_index` (`id\_trajet`),

  KEY `covoiturage\_\_trajet\_id\_vehicule\_index` (`id\_vehicule`),

  KEY `covoiturage\_\_trajet\_code\_employe\_index` (`code\_employe`),

  CONSTRAINT `covoiturage\_\_trajet\_code\_employe\_foreign` FOREIGN KEY (`code\_employe`) REFERENCES `employe\_\_employe` (`code\_employe`),

  CONSTRAINT `covoiturage\_\_trajet\_id\_vehicule\_foreign` FOREIGN KEY (`id\_vehicule`) REFERENCES `covoiturage\_\_vehicule` (`id\_vehicule`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci;

LOCK TABLES `covoiturage\_\_trajet` WRITE;

UNLOCK TABLES;

DROP TABLE IF EXISTS `covoiturage\_\_vehicule`;

CREATE TABLE `covoiturage\_\_vehicule` (

  `id\_vehicule` bigint(20) unsigned NOT NULL AUTO\_INCREMENT,

  `immatriculation` varchar(15) COLLATE utf8mb4\_unicode\_ci NOT NULL,

  `marque` varchar(15) COLLATE utf8mb4\_unicode\_ci NOT NULL,

  `model` varchar(15) COLLATE utf8mb4\_unicode\_ci NOT NULL,

  `annee\_model` year(4) NOT NULL,

  `type\_vehicule` enum('perso','service') COLLATE utf8mb4\_unicode\_ci NOT NULL,

  `id\_agence` bigint(20) unsigned DEFAULT NULL,

  `code\_employe` bigint(20) unsigned DEFAULT NULL,

  `created\_at` timestamp NULL DEFAULT NULL,

  `updated\_at` timestamp NULL DEFAULT NULL,

  PRIMARY KEY (`id\_vehicule`),

  KEY `covoiturage\_\_vehicule\_id\_vehicule\_index` (`id\_vehicule`),

  KEY `covoiturage\_\_vehicule\_id\_agence\_index` (`id\_agence`),

  KEY `covoiturage\_\_vehicule\_code\_employe\_index` (`code\_employe`),

  CONSTRAINT `covoiturage\_\_vehicule\_code\_employe\_foreign` FOREIGN KEY (`code\_employe`) REFERENCES `employe\_\_employe` (`code\_employe`),

  CONSTRAINT `covoiturage\_\_vehicule\_id\_agence\_foreign` FOREIGN KEY (`id\_agence`) REFERENCES `salle\_\_agence` (`id\_agence`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci;

LOCK TABLES `covoiturage\_\_vehicule` WRITE;

UNLOCK TABLES;

DROP TABLE IF EXISTS `employe\_\_employe`;

CREATE TABLE `employe\_\_employe` (

  `code\_employe` bigint(20) unsigned NOT NULL AUTO\_INCREMENT,

  `prenom` varchar(30) COLLATE utf8mb4\_unicode\_ci NOT NULL,

  `nom` varchar(30) COLLATE utf8mb4\_unicode\_ci NOT NULL,

  `utilisateur` varchar(50) COLLATE utf8mb4\_unicode\_ci NOT NULL,

  `mot\_de\_passe` varchar(255) COLLATE utf8mb4\_unicode\_ci NOT NULL,

  `email` varchar(100) COLLATE utf8mb4\_unicode\_ci NOT NULL,

  `date\_naissance` date NOT NULL,

  `date\_embauche` date NOT NULL,

  `derniere\_connexion` datetime DEFAULT NULL,

  `id\_agence` bigint(20) unsigned NOT NULL,

  `code\_fonction` bigint(20) unsigned NOT NULL,

  `created\_at` timestamp NULL DEFAULT NULL,

  `updated\_at` timestamp NULL DEFAULT NULL,

  PRIMARY KEY (`code\_employe`),

  KEY `employe\_\_employe\_code\_employe\_index` (`code\_employe`),

  KEY `employe\_\_employe\_id\_agence\_index` (`id\_agence`),

  KEY `employe\_\_employe\_code\_fonction\_index` (`code\_fonction`),

  CONSTRAINT `employe\_\_employe\_code\_fonction\_foreign` FOREIGN KEY (`code\_fonction`) REFERENCES `employe\_\_fonction` (`code\_fonction`),

  CONSTRAINT `employe\_\_employe\_id\_agence\_foreign` FOREIGN KEY (`id\_agence`) REFERENCES `salle\_\_agence` (`id\_agence`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci;

LOCK TABLES `employe\_\_employe` WRITE;

UNLOCK TABLES;

DROP TABLE IF EXISTS `employe\_\_fonction`;

CREATE TABLE `employe\_\_fonction` (

  `code\_fonction` bigint(20) unsigned NOT NULL AUTO\_INCREMENT,

  `nom\_fonction` varchar(50) COLLATE utf8mb4\_unicode\_ci NOT NULL,

  `created\_at` timestamp NULL DEFAULT NULL,

  `updated\_at` timestamp NULL DEFAULT NULL,

  PRIMARY KEY (`code\_fonction`),

  KEY `employe\_\_fonction\_code\_fonction\_index` (`code\_fonction`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci;

LOCK TABLES `employe\_\_fonction` WRITE;

UNLOCK TABLES;

DROP TABLE IF EXISTS `frais\_\_frais`;

CREATE TABLE `frais\_\_frais` (

  `id\_frais` bigint(20) unsigned NOT NULL AUTO\_INCREMENT,

  `montant\_total` decimal(16,2) NOT NULL,

  `date\_frais` date NOT NULL,

  `type\_forfait` enum('forfait','horforfait') COLLATE utf8mb4\_unicode\_ci NOT NULL,

  `horsforfait\_libelle` varchar(50) COLLATE utf8mb4\_unicode\_ci DEFAULT NULL,

  `forfait\_quantite` int(11) DEFAULT NULL,

  `commentaire` varchar(255) COLLATE utf8mb4\_unicode\_ci DEFAULT NULL,

  `appartenance\_mois` char(2) COLLATE utf8mb4\_unicode\_ci NOT NULL,

  `appartenance\_annee` year(4) NOT NULL,

  `id\_nature` bigint(20) unsigned DEFAULT NULL,

  `code\_situation` bigint(20) unsigned NOT NULL,

  `code\_employe\_comptable` bigint(20) unsigned DEFAULT NULL,

  `id\_visite` bigint(20) unsigned NOT NULL,

  `created\_at` timestamp NULL DEFAULT NULL,

  `updated\_at` timestamp NULL DEFAULT NULL,

  PRIMARY KEY (`id\_frais`),

  KEY `frais\_\_frais\_id\_frais\_index` (`id\_frais`),

  KEY `frais\_\_frais\_id\_nature\_index` (`id\_nature`),

  KEY `frais\_\_frais\_code\_situation\_index` (`code\_situation`),

  KEY `frais\_\_frais\_code\_employe\_comptable\_index` (`code\_employe\_comptable`),

  KEY `frais\_\_frais\_id\_visite\_index` (`id\_visite`),

  CONSTRAINT `frais\_\_frais\_code\_employe\_comptable\_foreign` FOREIGN KEY (`code\_employe\_comptable`) REFERENCES `employe\_\_employe` (`code\_employe`),

  CONSTRAINT `frais\_\_frais\_code\_situation\_foreign` FOREIGN KEY (`code\_situation`) REFERENCES `frais\_\_situation\_validation` (`code\_situation`),

  CONSTRAINT `frais\_\_frais\_id\_nature\_foreign` FOREIGN KEY (`id\_nature`) REFERENCES `frais\_\_nature` (`id\_nature`),

  CONSTRAINT `frais\_\_frais\_id\_visite\_foreign` FOREIGN KEY (`id\_visite`) REFERENCES `visite\_\_visite` (`id\_visite`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci;

LOCK TABLES `frais\_\_frais` WRITE;

UNLOCK TABLES;

DROP TABLE IF EXISTS `frais\_\_justificative`;

CREATE TABLE `frais\_\_justificative` (

  `id\_justif` bigint(20) unsigned NOT NULL AUTO\_INCREMENT,

  `justif\_chemin` varchar(255) COLLATE utf8mb4\_unicode\_ci NOT NULL,

  `justif\_extension` varchar(5) COLLATE utf8mb4\_unicode\_ci NOT NULL,

  `justif\_mime` varchar(20) COLLATE utf8mb4\_unicode\_ci NOT NULL,

  `id\_frais` bigint(20) unsigned NOT NULL,

  `created\_at` timestamp NULL DEFAULT NULL,

  `updated\_at` timestamp NULL DEFAULT NULL,

  PRIMARY KEY (`id\_justif`),

  KEY `frais\_\_justificative\_id\_justif\_index` (`id\_justif`),

  KEY `frais\_\_justificative\_id\_frais\_index` (`id\_frais`),

  CONSTRAINT `frais\_\_justificative\_id\_frais\_foreign` FOREIGN KEY (`id\_frais`) REFERENCES `frais\_\_frais` (`id\_frais`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci;

LOCK TABLES `frais\_\_justificative` WRITE;

UNLOCK TABLES;

DROP TABLE IF EXISTS `frais\_\_nature`;

CREATE TABLE `frais\_\_nature` (

  `id\_nature` bigint(20) unsigned NOT NULL AUTO\_INCREMENT,

  `intitule\_frais` varchar(50) COLLATE utf8mb4\_unicode\_ci NOT NULL,

  `montant\_forfait` decimal(16,2) NOT NULL,

  `created\_at` timestamp NULL DEFAULT NULL,

  `updated\_at` timestamp NULL DEFAULT NULL,

  PRIMARY KEY (`id\_nature`),

  KEY `frais\_\_nature\_id\_nature\_index` (`id\_nature`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci;

LOCK TABLES `frais\_\_nature` WRITE;

UNLOCK TABLES;

DROP TABLE IF EXISTS `frais\_\_situation\_validation`;

CREATE TABLE `frais\_\_situation\_validation` (

  `code\_situation` bigint(20) unsigned NOT NULL AUTO\_INCREMENT,

  `libelle\_situation` varchar(50) COLLATE utf8mb4\_unicode\_ci NOT NULL,

  `created\_at` timestamp NULL DEFAULT NULL,

  `updated\_at` timestamp NULL DEFAULT NULL,

  PRIMARY KEY (`code\_situation`),

  KEY `frais\_\_situation\_validation\_code\_situation\_index` (`code\_situation`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci;

LOCK TABLES `frais\_\_situation\_validation` WRITE;

UNLOCK TABLES;

DROP TABLE IF EXISTS `parametrage\_\_departement`;

CREATE TABLE `parametrage\_\_departement` (

  `departement\_id` bigint(20) unsigned NOT NULL AUTO\_INCREMENT,

  `nom\_departement` varchar(50) COLLATE utf8mb4\_unicode\_ci NOT NULL,

  `region\_id` bigint(20) unsigned NOT NULL,

  `created\_at` timestamp NULL DEFAULT NULL,

  `updated\_at` timestamp NULL DEFAULT NULL,

  PRIMARY KEY (`departement\_id`),

  KEY `parametrage\_\_departement\_departement\_id\_index` (`departement\_id`),

  KEY `parametrage\_\_departement\_region\_id\_index` (`region\_id`),

  CONSTRAINT `parametrage\_\_departement\_region\_id\_foreign` FOREIGN KEY (`region\_id`) REFERENCES `parametrage\_\_region` (`region\_id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci;

LOCK TABLES `parametrage\_\_departement` WRITE;

UNLOCK TABLES;

DROP TABLE IF EXISTS `parametrage\_\_region`;

CREATE TABLE `parametrage\_\_region` (

  `region\_id` bigint(20) unsigned NOT NULL AUTO\_INCREMENT,

  `nom\_region` varchar(50) COLLATE utf8mb4\_unicode\_ci NOT NULL,

  `created\_at` timestamp NULL DEFAULT NULL,

  `updated\_at` timestamp NULL DEFAULT NULL,

  PRIMARY KEY (`region\_id`),

  KEY `parametrage\_\_region\_region\_id\_index` (`region\_id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci;

LOCK TABLES `parametrage\_\_region` WRITE;

UNLOCK TABLES;

DROP TABLE IF EXISTS `parametrage\_\_ville`;

CREATE TABLE `parametrage\_\_ville` (

  `id\_ville` bigint(20) unsigned NOT NULL AUTO\_INCREMENT,

  `nom` varchar(50) COLLATE utf8mb4\_unicode\_ci NOT NULL,

  `code\_postal` char(5) COLLATE utf8mb4\_unicode\_ci NOT NULL,

  `ville\_longitude` decimal(9,6) NOT NULL,

  `ville\_latitude` decimal(8,6) NOT NULL,

  `departement\_id` bigint(20) unsigned NOT NULL,

  `created\_at` timestamp NULL DEFAULT NULL,

  `updated\_at` timestamp NULL DEFAULT NULL,

  PRIMARY KEY (`id\_ville`),

  KEY `parametrage\_\_ville\_id\_ville\_index` (`id\_ville`),

  KEY `parametrage\_\_ville\_nom\_index` (`nom`),

  KEY `parametrage\_\_ville\_code\_postal\_index` (`code\_postal`),

  KEY `parametrage\_\_ville\_departement\_id\_index` (`departement\_id`),

  CONSTRAINT `parametrage\_\_ville\_departement\_id\_foreign` FOREIGN KEY (`departement\_id`) REFERENCES `parametrage\_\_departement` (`departement\_id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci;

LOCK TABLES `parametrage\_\_ville` WRITE;

UNLOCK TABLES;

DROP TABLE IF EXISTS `salle\_\_agence`;

CREATE TABLE `salle\_\_agence` (

  `id\_agence` bigint(20) unsigned NOT NULL AUTO\_INCREMENT,

  `nom\_agence` varchar(50) COLLATE utf8mb4\_unicode\_ci NOT NULL,

  `id\_ville` bigint(20) unsigned NOT NULL,

  `created\_at` timestamp NULL DEFAULT NULL,

  `updated\_at` timestamp NULL DEFAULT NULL,

  PRIMARY KEY (`id\_agence`),

  KEY `salle\_\_agence\_id\_agence\_index` (`id\_agence`),

  KEY `salle\_\_agence\_id\_ville\_index` (`id\_ville`),

  CONSTRAINT `salle\_\_agence\_id\_ville\_foreign` FOREIGN KEY (`id\_ville`) REFERENCES `parametrage\_\_ville` (`id\_ville`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci;

LOCK TABLES `salle\_\_agence` WRITE;

UNLOCK TABLES;

DROP TABLE IF EXISTS `salle\_\_batiment`;

CREATE TABLE `salle\_\_batiment` (

  `id\_batiment` bigint(20) unsigned NOT NULL AUTO\_INCREMENT,

  `nom\_batiment` varchar(50) COLLATE utf8mb4\_unicode\_ci NOT NULL,

  `id\_agence` bigint(20) unsigned NOT NULL,

  `created\_at` timestamp NULL DEFAULT NULL,

  `updated\_at` timestamp NULL DEFAULT NULL,

  PRIMARY KEY (`id\_batiment`),

  KEY `salle\_\_batiment\_id\_batiment\_index` (`id\_batiment`),

  KEY `salle\_\_batiment\_id\_agence\_index` (`id\_agence`),

  CONSTRAINT `salle\_\_batiment\_id\_agence\_foreign` FOREIGN KEY (`id\_agence`) REFERENCES `salle\_\_agence` (`id\_agence`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci;

LOCK TABLES `salle\_\_batiment` WRITE;

UNLOCK TABLES;

DROP TABLE IF EXISTS `salle\_\_extra`;

CREATE TABLE `salle\_\_extra` (

  `id\_extra` bigint(20) unsigned NOT NULL AUTO\_INCREMENT,

  `libelle\_extra` varchar(100) COLLATE utf8mb4\_unicode\_ci NOT NULL,

  `created\_at` timestamp NULL DEFAULT NULL,

  `updated\_at` timestamp NULL DEFAULT NULL,

  PRIMARY KEY (`id\_extra`),

  KEY `salle\_\_extra\_id\_extra\_index` (`id\_extra`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci;

LOCK TABLES `salle\_\_extra` WRITE;

UNLOCK TABLES;

DROP TABLE IF EXISTS `salle\_\_materiel\_type`;

CREATE TABLE `salle\_\_materiel\_type` (

  `id\_materiel` bigint(20) unsigned NOT NULL AUTO\_INCREMENT,

  `nom\_materiel` varchar(255) COLLATE utf8mb4\_unicode\_ci NOT NULL,

  `created\_at` timestamp NULL DEFAULT NULL,

  `updated\_at` timestamp NULL DEFAULT NULL,

  PRIMARY KEY (`id\_materiel`),

  KEY `salle\_\_materiel\_type\_id\_materiel\_index` (`id\_materiel`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci;

LOCK TABLES `salle\_\_materiel\_type` WRITE;

UNLOCK TABLES;

DROP TABLE IF EXISTS `salle\_\_posession\_materiel`;

CREATE TABLE `salle\_\_posession\_materiel` (

  `id\_salle` bigint(20) unsigned NOT NULL,

  `id\_materiel` bigint(20) unsigned NOT NULL,

  `created\_at` timestamp NULL DEFAULT NULL,

  `updated\_at` timestamp NULL DEFAULT NULL,

  PRIMARY KEY (`id\_salle`,`id\_materiel`),

  KEY `salle\_\_posession\_materiel\_id\_salle\_index` (`id\_salle`),

  KEY `salle\_\_posession\_materiel\_id\_materiel\_index` (`id\_materiel`),

  CONSTRAINT `salle\_\_posession\_materiel\_id\_materiel\_foreign` FOREIGN KEY (`id\_materiel`) REFERENCES `salle\_\_materiel\_type` (`id\_materiel`),

  CONSTRAINT `salle\_\_posession\_materiel\_id\_salle\_foreign` FOREIGN KEY (`id\_salle`) REFERENCES `salle\_\_salle` (`id\_salle`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci;

LOCK TABLES `salle\_\_posession\_materiel` WRITE;

UNLOCK TABLES;

DROP TABLE IF EXISTS `salle\_\_reservation`;

CREATE TABLE `salle\_\_reservation` (

  `id\_reservation` bigint(20) unsigned NOT NULL AUTO\_INCREMENT,

  `date\_debut\_reservation` datetime NOT NULL,

  `date\_fin\_reservation` datetime NOT NULL,

  `nombre\_de\_personnes` int(11) NOT NULL,

  `code\_employe` bigint(20) unsigned NOT NULL,

  `id\_salle` bigint(20) unsigned NOT NULL,

  `id\_extra` bigint(20) unsigned DEFAULT NULL,

  `created\_at` timestamp NULL DEFAULT NULL,

  `updated\_at` timestamp NULL DEFAULT NULL,

  PRIMARY KEY (`id\_reservation`),

  KEY `salle\_\_reservation\_id\_reservation\_index` (`id\_reservation`),

  KEY `salle\_\_reservation\_code\_employe\_index` (`code\_employe`),

  KEY `salle\_\_reservation\_id\_salle\_index` (`id\_salle`),

  KEY `salle\_\_reservation\_id\_extra\_index` (`id\_extra`),

  CONSTRAINT `salle\_\_reservation\_code\_employe\_foreign` FOREIGN KEY (`code\_employe`) REFERENCES `employe\_\_employe` (`code\_employe`),

  CONSTRAINT `salle\_\_reservation\_id\_extra\_foreign` FOREIGN KEY (`id\_extra`) REFERENCES `salle\_\_extra` (`id\_extra`),

  CONSTRAINT `salle\_\_reservation\_id\_salle\_foreign` FOREIGN KEY (`id\_salle`) REFERENCES `salle\_\_salle` (`id\_salle`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci;

LOCK TABLES `salle\_\_reservation` WRITE;

UNLOCK TABLES;

DROP TABLE IF EXISTS `salle\_\_salle`;

CREATE TABLE `salle\_\_salle` (

  `id\_salle` bigint(20) unsigned NOT NULL AUTO\_INCREMENT,

  `nom\_salle` varchar(20) COLLATE utf8mb4\_unicode\_ci NOT NULL,

  `id\_batiment` bigint(20) unsigned NOT NULL,

  `created\_at` timestamp NULL DEFAULT NULL,

  `updated\_at` timestamp NULL DEFAULT NULL,

  PRIMARY KEY (`id\_salle`),

  KEY `salle\_\_salle\_id\_salle\_index` (`id\_salle`),

  KEY `salle\_\_salle\_id\_batiment\_index` (`id\_batiment`),

  CONSTRAINT `salle\_\_salle\_id\_batiment\_foreign` FOREIGN KEY (`id\_batiment`) REFERENCES `salle\_\_batiment` (`id\_batiment`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci;

LOCK TABLES `salle\_\_salle` WRITE;

UNLOCK TABLES;

DROP TABLE IF EXISTS `visite\_\_famille\_medicament`;

CREATE TABLE `visite\_\_famille\_medicament` (

  `id\_famille` bigint(20) unsigned NOT NULL AUTO\_INCREMENT,

  `nom\_famille` varchar(30) COLLATE utf8mb4\_unicode\_ci NOT NULL,

  `created\_at` timestamp NULL DEFAULT NULL,

  `updated\_at` timestamp NULL DEFAULT NULL,

  PRIMARY KEY (`id\_famille`),

  KEY `visite\_\_famille\_medicament\_id\_famille\_index` (`id\_famille`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci;

LOCK TABLES `visite\_\_famille\_medicament` WRITE;

UNLOCK TABLES;

DROP TABLE IF EXISTS `visite\_\_medicament`;

CREATE TABLE `visite\_\_medicament` (

  `id\_medicament` bigint(20) unsigned NOT NULL AUTO\_INCREMENT,

  `nom\_medicament` varchar(30) COLLATE utf8mb4\_unicode\_ci NOT NULL,

  `photo\_medicament` varchar(255) COLLATE utf8mb4\_unicode\_ci NOT NULL,

  `id\_famille` bigint(20) unsigned NOT NULL,

  `created\_at` timestamp NULL DEFAULT NULL,

  `updated\_at` timestamp NULL DEFAULT NULL,

  PRIMARY KEY (`id\_medicament`),

  KEY `visite\_\_medicament\_id\_medicament\_index` (`id\_medicament`),

  KEY `visite\_\_medicament\_id\_famille\_index` (`id\_famille`),

  CONSTRAINT `visite\_\_medicament\_id\_famille\_foreign` FOREIGN KEY (`id\_famille`) REFERENCES `visite\_\_famille\_medicament` (`id\_famille`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci;

LOCK TABLES `visite\_\_medicament` WRITE;

UNLOCK TABLES;

DROP TABLE IF EXISTS `visite\_\_praticien`;

CREATE TABLE `visite\_\_praticien` (

  `id\_praticien` bigint(20) unsigned NOT NULL AUTO\_INCREMENT,

  `raison\_sociale` varchar(50) COLLATE utf8mb4\_unicode\_ci NOT NULL,

  `prenom` varchar(30) COLLATE utf8mb4\_unicode\_ci NOT NULL,

  `nom` varchar(30) COLLATE utf8mb4\_unicode\_ci NOT NULL,

  `adresse` varchar(100) COLLATE utf8mb4\_unicode\_ci NOT NULL,

  `id\_ville` bigint(20) unsigned NOT NULL,

  `created\_at` timestamp NULL DEFAULT NULL,

  `updated\_at` timestamp NULL DEFAULT NULL,

  PRIMARY KEY (`id\_praticien`),

  KEY `visite\_\_praticien\_id\_praticien\_index` (`id\_praticien`),

  KEY `visite\_\_praticien\_id\_ville\_index` (`id\_ville`),

  CONSTRAINT `visite\_\_praticien\_id\_ville\_foreign` FOREIGN KEY (`id\_ville`) REFERENCES `parametrage\_\_ville` (`id\_ville`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci;

LOCK TABLES `visite\_\_praticien` WRITE;

UNLOCK TABLES;

DROP TABLE IF EXISTS `visite\_\_presentation\_medicament`;

CREATE TABLE `visite\_\_presentation\_medicament` (

  `id\_visite` bigint(20) unsigned NOT NULL,

  `id\_medicament` bigint(20) unsigned NOT NULL,

  `created\_at` timestamp NULL DEFAULT NULL,

  `updated\_at` timestamp NULL DEFAULT NULL,

  PRIMARY KEY (`id\_visite`,`id\_medicament`),

  KEY `visite\_\_presentation\_medicament\_id\_visite\_index` (`id\_visite`),

  KEY `visite\_\_presentation\_medicament\_id\_medicament\_index` (`id\_medicament`),

  CONSTRAINT `visite\_\_presentation\_medicament\_id\_medicament\_foreign` FOREIGN KEY (`id\_medicament`) REFERENCES `visite\_\_medicament` (`id\_medicament`),

  CONSTRAINT `visite\_\_presentation\_medicament\_id\_visite\_foreign` FOREIGN KEY (`id\_visite`) REFERENCES `visite\_\_visite` (`id\_visite`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci;

LOCK TABLES `visite\_\_presentation\_medicament` WRITE;

UNLOCK TABLES;

DROP TABLE IF EXISTS `visite\_\_visite`;

CREATE TABLE `visite\_\_visite` (

  `id\_visite` bigint(20) unsigned NOT NULL AUTO\_INCREMENT,

  `date\_debut\_visite` date NOT NULL,

  `date\_fin\_visite` date NOT NULL,

  `rapport` longtext COLLATE utf8mb4\_unicode\_ci NOT NULL,

  `id\_praticien` bigint(20) unsigned NOT NULL,

  `code\_employe` bigint(20) unsigned NOT NULL,

  `created\_at` timestamp NULL DEFAULT NULL,

  `updated\_at` timestamp NULL DEFAULT NULL,

  PRIMARY KEY (`id\_visite`),

  KEY `visite\_\_visite\_id\_visite\_index` (`id\_visite`),

  KEY `visite\_\_visite\_id\_praticien\_index` (`id\_praticien`),

  KEY `visite\_\_visite\_code\_employe\_index` (`code\_employe`),

  CONSTRAINT `visite\_\_visite\_code\_employe\_foreign` FOREIGN KEY (`code\_employe`) REFERENCES `employe\_\_employe` (`code\_employe`),

  CONSTRAINT `visite\_\_visite\_id\_praticien\_foreign` FOREIGN KEY (`id\_praticien`) REFERENCES `visite\_\_praticien` (`id\_praticien`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci;

LOCK TABLES `visite\_\_visite` WRITE;

UNLOCK TABLES;

# Modélisation UML