

INF3710 –Fichiers et Bases de données

Hiver 2019

TP No. 5

Groupe 3

1897402 – Marilou Farmer

1907605 – Antoine Lamontagne

Soumis à : Nikolay Radoev

14 Avril 2019

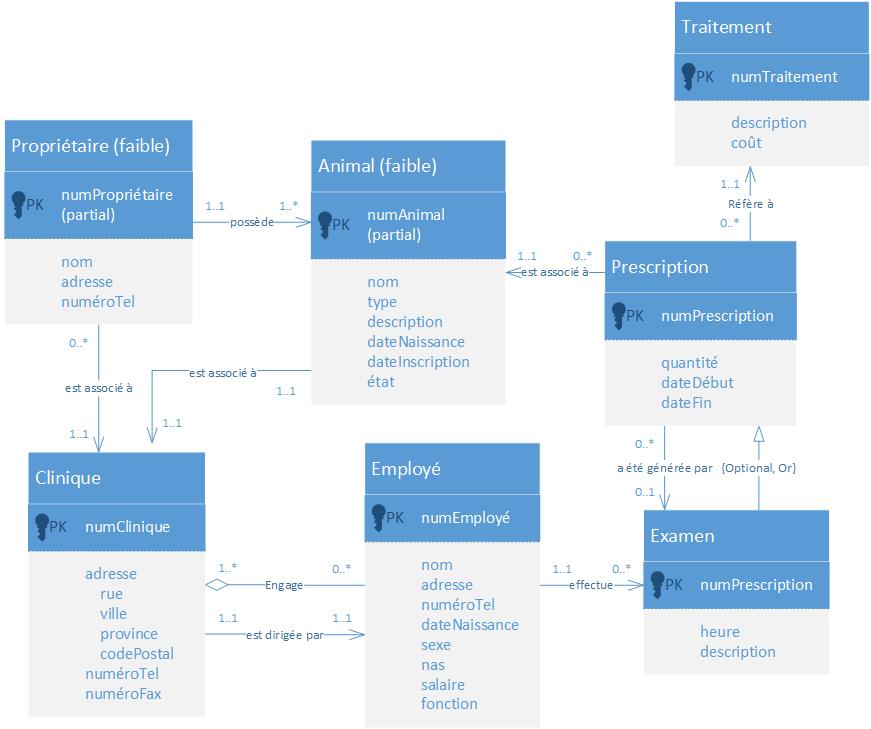
# Introduction

Le projet suivant est une application web pour la compagnie fictive VetoSansFrontieres, qui fournit des soins de santé privés pour animaux domestiques à travers le Canada. L’objectif de l’application est d’augmenter le niveau de communication des différentes cliniques. Les différents propriétaires ainsi que leurs animaux, les traitements, les employés et les cliniques sont les données qui peuvent être enregistrées grâce à l’application. Celle-ci est composée d’une application client, d’une application serveur et d’une base de données. Les technologies principales employées sont Angular, Express et PostgreSQL.

# Manuel de l’utilisateur

Afin d’assurer un bon fonctionnement de l’application, veuillez vous référer au fichier README.md et veuillez suivre les étapes dans l’ordre selon dans lequel elles sont présentées.

# Le modèle conceptuel UML



## Hypothèses

1. L'attribut nom pour les employés et les propriétaires contient le prénom, le deuxième nom et le nom de famille.
2. Un examen est un type de traitement. Il peut être prescrit à la fin d’un examen précédent par un vétérinaire ou bien avoir été enregistré sans examen par une secrétaire. Dans le second cas, l’attribut « numExamen » de prescription est null.
3. La date d’un examen est obtenue avec la date de début de la prescription ou avec la date de fin de la prescription associée.
4. Dans la description du problème, il est indiqué qu’on doit savoir le nom du vétérinaire pour chaque examen. Nous avons plutôt inclus une clé étrangère qui nous permet d’accéder au nom et à tous les attributs du vétérinaire.
5. Pour gérer une clinique, la fonction de l’employé doit absolument être « gestionnaire ».
6. Pour effectuer un examen, la fonction de l’employé doit absolument être « vétérinaire ».
7. Seule l’adresse de la clinique est un attribut composé, comme spécifié dans l’énoncé.

# Le modèle relationnel en syntaxe abstraite

Employe(numEmploye, nom, adresse, numTel, dateNaissance, sexe, nas, salaire, fonction)

PK(numEmploye)

Clinique(numClinique, adresse, numTel, numFax, numGestionnaire)

PK(numClinique)

FK(numGestionnaire) REFERENCES Employe(numEmploye)

Embauche(numEmploye, numClinique)

PK(numEmploye, numClinique)

FK(numEmploye) REFERENCES Employe(numEmploye)

FK(numClinique) REFERENCES Clinique(numClinique)

Proprietaire(numProprietaire numClinique, nom, adresse, numTel)

PK(numProprietaire, numClinique)

FK(numClinique) REFERENCES Clinique(numClinique)

Animal(numAnimal, numProprietaire, numClinique, nom, type, description, dateNaissance, dateInscription, etat)

PK(numAnimal, numClinique)

FK(numProprietaire) REFERENCES Proprietaire(numProprietaire)

FK(numClinique) REFERENCES Clinique(numClinique)

Traitement(numTraitement, description, cout)

PK(numTraitement)

Prescription(numPrescription, numAnimal, numTraitement, numExamen, quantite, dateDebut, dateFin)

PK(numPrescription)

FK(numAnimal) REFERENCES Animal(numAnimal)

FK(numTraitement) REFERENCES Traitement(numTraitement)

FK(numExamen) REFERENCES Examen(numExamen)

Examen(numPrescription, heure, description, numVeterinaire)

PK(numPrescription)

FK(numPrescription) REFERENCES Prescription(numPrescription)

FK(numVeterinaire) REFERENCES Employe(numEmploye)

# Dépendances fonctionnelles et explication de la forme normale

## Fonctionnalité bonus

# Présentation de l’application développée