

**INF3710 –Fichiers et Bases de données**

**Hiver 2019**

**TP No. 5**

**Groupe 4**

1340155 – Alassane Maiga

1862313 – Mohamed Esseddik BENYAHIA

**Soumis à : Manel Grinchi.**

**14 Avril 2019**

Table des matières

[I) Introduction 2](#_Toc6138412)

[II) Le modèle conceptuel UML incluant les hypothèses et commentaires 2](#_Toc6138413)

[III) Le modèle relationnel en syntaxe abstraite 3](#_Toc6138414)

[IV) Les dépendances fonctionnelles et l’explication de la forme normale de 5](#_Toc6138415)

[V) Création de la base de donnée PostgreSQL 6](#_Toc6138416)

[VI) Insertion des données 6](#_Toc6138417)

[VII) Création des requêtes 6](#_Toc6138418)

[VIII) Création de l’application web 8](#_Toc6138419)

## I) Introduction

Ceci est notre rapport du projet du cours de bases de données inf3710, session d’hiver 2019. Le projet consiste à développer un système de bases de données centralisées pour la gestion et la communication entre les cliniques d’une organisation appelée VetoSansFrontieres. De plus, il nous a été demandé de créer une application web permettant l’interrogation de la base de données.

Ce projet est important par le fait qu’il nous permet de revoir les principales notions de base vues dans le cours en les mettant en pratique sur un exemple réel. À cela s’ajouter le fait que le projet nous permet de passer par toutes les étapes de création d’un système de base de données, en allant de l’étape de conception jusqu’à l’étape de création des requêtes de manipulation de la table.

Dans ce qui suit nous détaillerons les différentes tâches qui nous ont été assignées dans le projet.

## II) Le modèle conceptuel UML incluant les hypothèses et commentaires

Après avoir lu et noté les besoins de l’organisme, nous avons proposé notre modèle entité association à l’aide du logiciel Visio et en utilisant la notation UML.

La figure 1 présente le modèle en question.

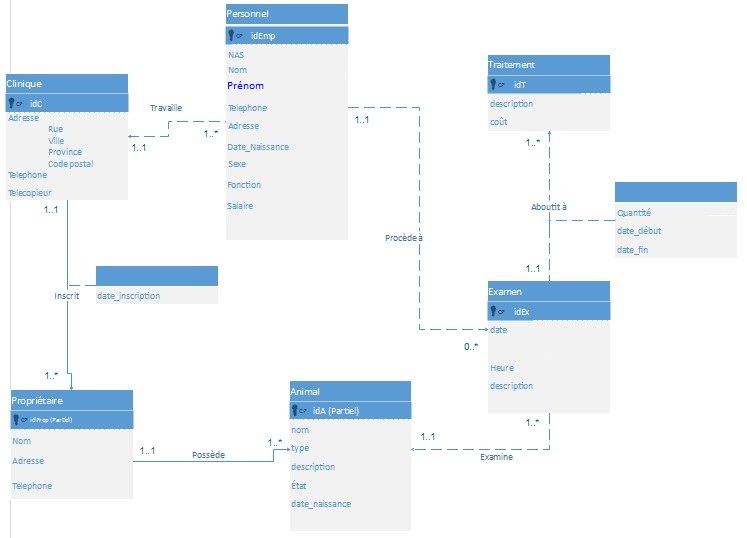


Figure 1: Modèle conceptuel.

Voici une description des hypothèses considérées dans notre modèle:

-) Un personnel travail uniquement dans une et une seule clinique

-) Un propriétaire ne peut s’inscrire que dans une et une seule clinique (c’est la première clinique qu’il visite).

-) Un animal ne peut être possédé que par un unique propriétaire.

-) La date d’inscription d’un animal correspond à la date d’inscription de son propriétaire.

-) Un examen aboutit à un ou plusieurs traitements, car dans le pire des cas un examen est considéré comme un traitement et il a un coût.

-) Nous avons choisi d’utiliser une seul entité pour tous les employés d’une clinique, en revanche, nous avons créé une fonction dans le dossier functions.sql qui impose à ce l’employé soit un vétérinaire pour pouvoir effectuer un examen.

## III) Le modèle relationnel en syntaxe abstraite

Voici la traduction du modèle conceptuel en modèle relationnel :

Clinic (idC, Rue, Ville, Province, Code postal, Téléphone, Télécopieur)

Primary Key(idC)

Employe (idEmp, idC, NAS, Nom, Prénom, Téléphone, Adresse, Date de Naissance, Sexe, fonction, salaire)

Primary Key(idEmp)

Foreign Key(idC) REFERENCES Clinic(idC)

Owner (idO, idC, nom, Adresse, Téléphone, date d’inscription)

Primary Key(idO, idC)

Foreign Key (idC) REFERENCES Clinic(idC)

Pet (idP, idO, idC, nom, specie, description, DoB, status)

Primary Key (idP, idO, idC)

Foreign Key (idO, idC) REFERENCES Owner(idO, idC)

Exam (idEx, idP, idO, idC, idEmp, Date, Heure, description)

Primary Key (idEx)

Foreign Key(idEmp) REFERENCES Employee(idEmp)

Foreign Key(idP, idO, idC) REFERENCES Pet(idP, idO, idC)

Treatment (idT, idEx, description, cost, quantity, startDate, endDate)

Primary Key(idT)

Foreign Key (idExam) REFERENCES PhysicalExam(idExam)

IV) Les dépendances fonctionnelles et la forme normale dela base de données

Voici notre stratégie pour la forme normal :

• Trouver le ou les attributs qui peuvent servir de clé primaire dans la table non normalisée

• Identifier les groupes qui se répètent pour la clé choisie et distribuer les valeurs non atomiques sur des lignes différentes

• Trouver le ou les attributs qui peuvent servir de clé primaire dans la table non normalisée

• Identifier les groupes qui se répètent pour la clé choisie et distribuer les valeurs non atomiques sur des lignes différentes

Par conséquence, on obtient :

IdC  ClinNom, CAdresse, CTelephone, CTelecopieur, numGestionnaire

idEmp  IdC, NAS, EmpPrenom, EmpNom, EmpAdresse, EmpTelephone, EmpDateNaissance, EmpSexe, fonction, salaire

NAS  EmpPrenom, EmpNom, EmpAdresse, EmpTelephone, EmpDateNaissance, EmpSexe

idowner, IdC  ProPrenom, ProNom, ProAdresse, ProNumTelephone

numAnimal. idowner, IdC  AnNom, typeAnimal, AnDescription, AnDateNaissance, AnDateInscription, etatActuel

idEx  dateExam, descResultats, numVeterinaire, numAnimal

idTr  TrDescription, TrCout

idTr, idEx  quantite, dateDebut, dateFin

## V) Création de la base de donnée PostgreSQL

## Voir fichier bdschema.sql.

## VI) Insertion des données

Voir fichier Data.sql

## VII) Création des requêtes

Voir fichier query.sql

## VIII) Création de l’application web

Notre application fonction parfaitement et répond à tous les exigences.

Nous seront très content si vous auriez un peu de temps pour la tester. Merci

Tous les réponses sont dans registration et medical-file