



INF3710 –Fichiers et Bases de données

Hiver 2021

TP No. [3]

Groupe [13]

1995039 – [Labib Bashir-Choudhury]

2012859– [Philippe Bonhomme]

Soumis à :Mohamed Ali Elakhrass

[Avril 17 2021]

Introduction:

Les bases de données sont utilisées dans plusieurs applications utilisées par les étudiants au quotidien. Ceux-ci sont très importants et demandent un peu plus d'attention. De ce fait, ce projet nous permet de regarder cela en profondeur, Effectivement, _cette application web permet à l'utilisateur de voir les données d'une clinique. Cette clinique s'appelle Veto Sans-frontière. Cette application a été rédigée sur Angular et utilise PostGreSQL comme base de données. Cette application consiste en trois onglets, un onglet qui représente les détails de chaque animal. Un deuxième onglet qui affiche tous les traitements d'un animal, et un autre qui affiche la facture.

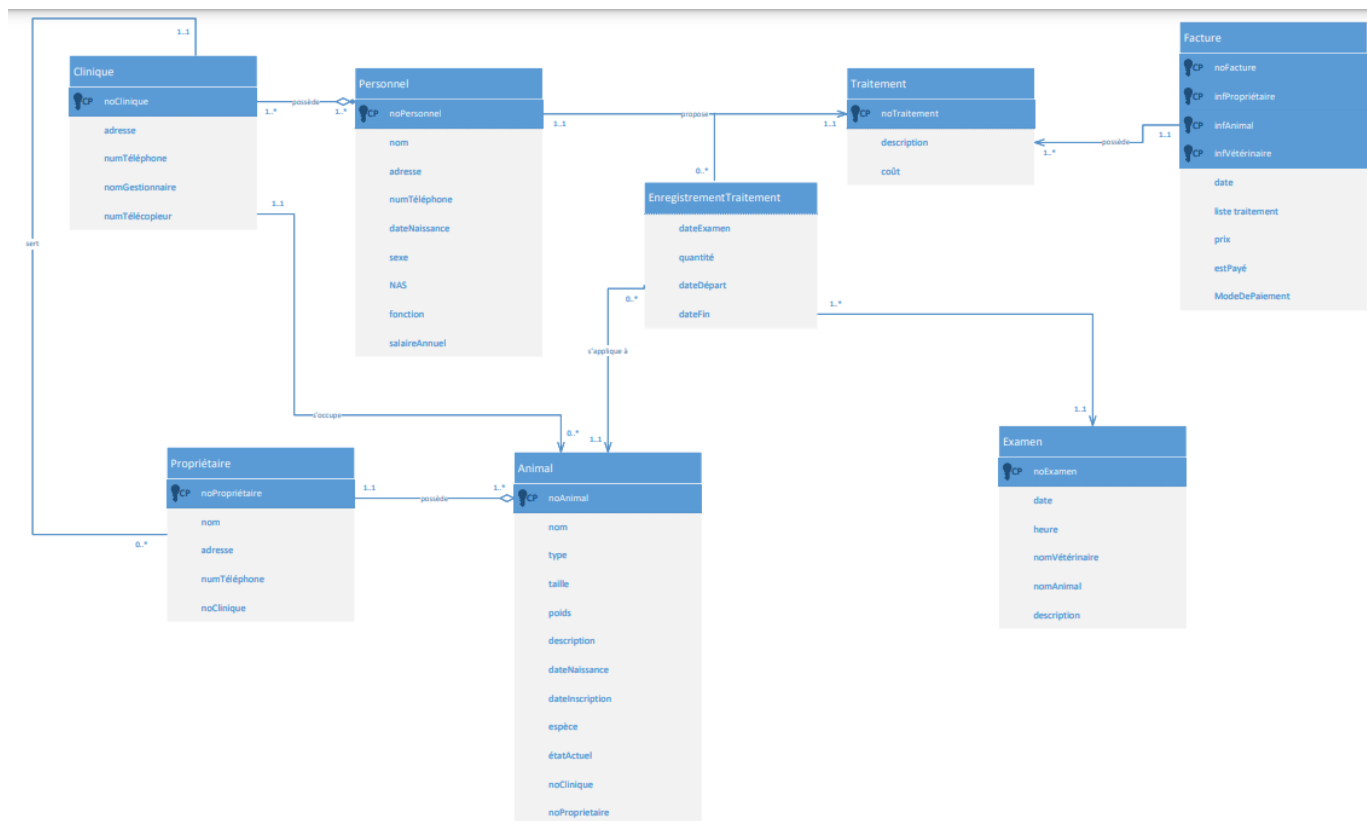


Table Clinique :

Clinique(NoClinique, adresse, numTéléphone, numTélécopieur ,nomGestionnaire)

PRIMARY KEY(NoClinique)

Dependence fonctionnel:

NoClinique → adresse, numTelephone, numTelecopieur, nomGestionnaire.

nomGestionnaire → NoClinique

Normalisation:

3NF, car il n'a pas de transitivité et tous les attributs sont déterminés par la clé primaire.

Table CliniqueRelPersonnel:

CliniqueRelPersonnel (NoClinique, adresse, numTéléphone, numTélécopieur)

PRIMARY KEY (NoClinique, noPersonnel)

FOREIGN KEY (noPersonnel) REFERENCES Personnel (noPersonnel)

FOREIGN KEY (NoClinique) REFERENCES Clinique (NoClinique)

Dépendance fonctionnelle

NoClinique, noPersonnel → adresse, numTelephone, numTelecopieur

NoClinique → adresse, numTelephone, numTelecopieur.

Il a une relation many:many entre clinique et personnel, donc il faut faire une table entre les deux qui a une copie des clés primaires de clinique et de personnels comme clés étrangères. Ses clés étrangères sont aussi les clés primaires de cette table.

Normalisation:

1NF car tous les attributs sont atomiques. Ce n'est pas 2NF car une clé candidate peut être assés pour démontrer la dépendance fonctionnelle.

Table Personnel:

Personnel (noPersonnel, nom, adresse, numTéléphone, dateNaissance, sexe, nas, fonction, salaireAnnuel)

PRIMARY KEY (noPersonnel)

Dépendance fonctionnelle

noPersonnel → (nom, adresse, numTelephone, dateNaissance, sexe, nas, fonction, salaireAnnuel)

nas → (nom, adresse, numTelephone, dateNaissance, sexe, nas, fonction, salaireAnnuel)

fonction \rightarrow (salaireAnnuel)

Normalisation:

Tous les attributs sont liés au clé primaire, donc il est en 2NF. Il n'est pas en 3NF car il a de la transitivité.

Table Facture:

Facture (noFacture, noPropriétaire, noAnimal, noVétérinaire, date, prix, estPayé, modeDePaiement)

PRIMARY KEY (NoFacture)

FOREIGN KEY (noVétérinaire) REFERENCES Personnel (noPersonnel)

FOREIGN KEY (noAnimal) REFERENCES Animal(noAnimal)

FOREIGN KEY (noPropriétaire) REFERENCES Proprietaire(noPropriétaire)

Facture a une relation *:1 avec Animal Personnel et Propriétaire. Alors on prend en clé étrangère la clé primaire de ces trois tables.

Dépendance fonctionnelle

noFacture \rightarrow noPropriétaire, noAnimal, noVétérinaire, date, prix, estPayé, modeDePaiement

Ceci est en 3NF car tous les attributs sont liés à la clé primaire et il n'a pas de transitivité.

Table Traitement :

Traitement(noTraitement, description, cout, noPersonnel, noFacture)

PRIMARY KEY (noTraitement)

FOREIGN KEY (noPersonnel) REFERENCES Personnel (noPersonnel)

FOREIGN KEY (noFacture) REFERENCES Facture(noFacture)

Traitement a une relation *:1 avec Personnel , donc il va prendre la clé primaire de Traitement comme clé étrangère pour sa table. Traitement a aussi une relation *:1 avec Facture, alors il prend encore la clé primaire pour clé étrangère.

Dépendance fonctionnelle

noTraitement → description, cout, noPersonnel, noFacture

noFacture → noPersonnel

2NF car ils sont tous reliés au clé primaire,, il n'est pas 3NF car il a de la transitivité.

Table EnregistrementTraitement:

EnregistrementTraitement(dateExamen, quantite, heureDepart, heureFin, noExamen, noAnimal, noVeterinaire)

PRIMARY KEY (noExamen, noAnimal, noVétérinaire)

FOREIGN KEY(noExamen) REFERENCES Examen(noExamen)

FOREIGN KEY(noAnimal) REFERENCES Animal(noAnimal)

FOREIGN KEY (noVeterinaire) REFERENCES Vétérinaire(Vétérinaire) .

Il a une relation 1:* entre EnregistrementTraitement et Animal, EnregistrementTraitement et Veterinaire, EnregistrementTraitement et Examen. Donc il faut prendre la clé primaire d'animal, de Vétérinaire et d'Examen et les mettre comme clés étrangères dans la table d'EnregistrementTraitement.

Dépendance fonctionnelle:

noExamen, noAnimal, noVétérinaire → dateExamen, quantite, heureDepart, heureFin

Normalisation:

3NF car tous les attributs sont reliés au clé primaire.

Table Examen:

Examen (NoExamen, date, heure, nomVeterinaire, nomAnimal, description)

PRIMARY KEY (NoExamen)

Dependence fonctionnelle:

NoExamen -> date, heure, nomVeterinaire, nomAnimal, description

Normalisation:

Ceci est un 2nf car chaque attribut dépend de la clé primaire.

Table Proprietaire

Proprietaire (noProprietaire, nom, adresse, numTelephone, noClinique)

PRIMARY KEY (noProprietaire)

FOREIGN KEY (noClinique) REFERENCES Clinique(noClinique)

Propriétaire a une relation *:1 avec clinique, donc pour cette raison propriétaire prend une copie de la clé primaire de clinique comme clé étrangère.

Dependence fonctionnelle:

noProprietaire -> nom, adresse, numTelephone, noClinique

Normalisation:

2NF car chaque attribut est relié au clé primaire.

Table Animal:

Animal(noAnimal,noClinique, noProprietaire, nom, type,taille, poids, description, dateNaissance, dateInscription, espece)

PRIMARY KEY (noAnimal)

FOREIGN KEY (noClinique) REFERENCES Clinique(noClinique)

FOREIGN KEY (noProprietaire) REFERENCES Proprietaire(noProprietaire)

Animal a une relation *:1 avec Clinique et propriétaire, donc il prend leur clé primaire comme clé étrangère.

Dépendance fonctionnelle

noAnimal → noClinique, noProprietaire, nom, type,taille, poids, description, dateNaissance, dateInscription, espece

espece → type

noProprietaire → noClinique

Normalisation:

2NF car chaque attribut est lié au clé primaire, mais il a de la transitivité à cause des deux dernières dépendances.

Table Historiques

Historiques (noPersonnel, noAnimal, noTraitement, noExamen)

PRIMARY KEY (noTraitement,noPersonnel,noAnimal,noExamen)

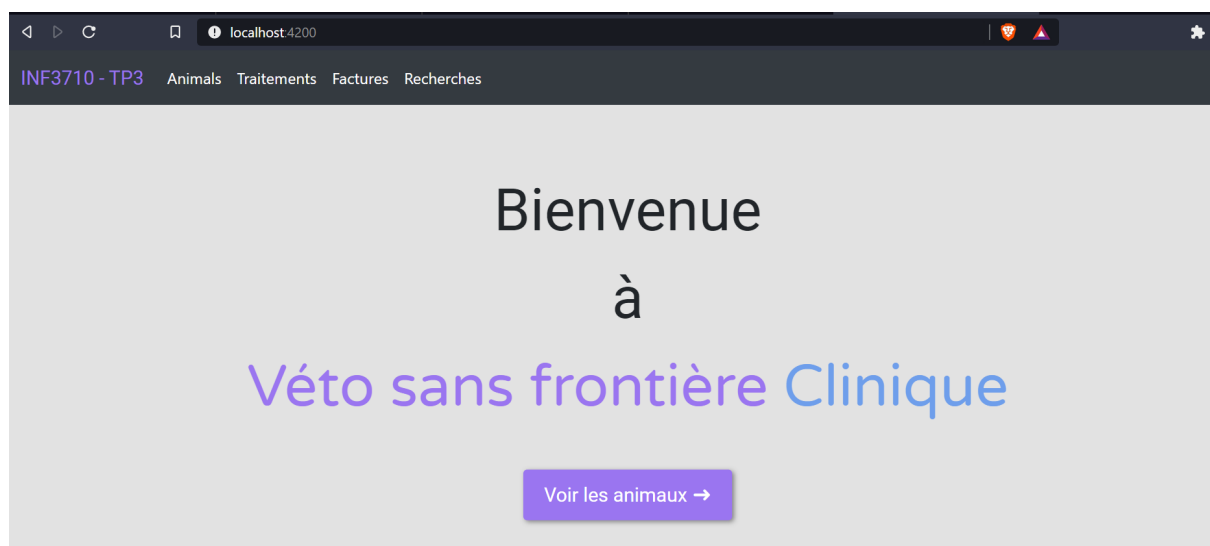
FOREIGN KEY (noPersonnel) REFERENCES Personnel (noPersonnel),

FOREIGN KEY (noTraitement) REFERENCES Traitement (noTraitement)

FOREIGN KEY (noAnimal) REFERENCES Animal(noAnimal)

FOREIGN KEY (noExamen) REFERENCES Examens(noExamen)

Cette table est utilisée pour le trigger. Pour l'historique, il est important de garder la clé primaire de tous les informations pertinents comme le personnel, animal, traitement et examen.



Voici la fenêtre d'introduction. Premièrement, il faut cliquer sur le bouton 'voir animaux', pour aller vers les données de la table animaux

Animal

NUMÉRO ANIMAL	NUMÉRO CLINIQUE	NUMÉRO PROPRIÉTAIRE	NOM	TYPE	TAILLE	ESPÈCE	POIDS	DESCRIPTION	DATE DE NAISSANCE	DATE D'INSCRIPTION	EST VACCINÉ?	ACTION
1	15	1	Grosse Patate	chat	20	Boxer	70	noir	2012-03-24	2015-03-20	true	 
2	15	2	Jujube	chien	65	Labrador	52	blanc	2018-05-19	2017-10-23	true	 
3	15	3	petit Coeur	chien	21	chiwouawoua	30	petit	2016-03-21	2016-04-20	false	 
Nouveau numero	15 ▾	1 ▾	Nouveau nom	Nouveau type	Nouveau taille	Nouveau especes	Nouveau poids	Nouveau description	Nouvelle Date de Naissance	Nouvelle Date d'inscription	Nouveau Vaccine	

+

Ici, on voit les informations de tous les animaux de toutes les cliniques. Le bouton mauve à droite en forme de poubelle supprime le tuple de la table. De plus, il est possible de modifier les éléments de la table en les modifiant à partir du champ de texte. Il est aussi possible d'ajouter un élément avec le bouton "+" en bas. Il y a des listes déroulantes pour les numéros de clinique et de propriétaire.

Traitement

Traitement 1 ▾

noAnimal	noTraitement	description	cout	nopersonnel	nofacture
1	1	anti-depresseur	20.4	3	1
1	2	cholesterol	183	3	1

Ensuite, pour les traitements, on peut sélectionner tous les traitements d'un animal spécifié dans la liste déroulante. La liste déroulante en haut représente les numéros d'animaux.

Facture

Animal 1

Numéro de Facture	Numéro de Propriétaire	Numéro de animal	Numéro de vétérinaire	date	prix	Est payé?	Mode de Paiement
1	1	1	3	2020-09-21T04:00:00.000Z	203.4	false	debit

Pour factures, il faut sélectionner les noAnimal avec la liste déroulante et l'application va afficher les informations de tous les factures dans la table.

Recherches

Nom de l'animal e

NUMÉRO ANIMAL	NUMÉRO CLINIQUE	NUMÉRO PROPRIÉTAIRE	NOM	TYPE	TAILLE	ESPÈCE	POIDS	DESCRIPTION	DATE DE NAISSANCE	DATE D'INSCRIPTION	EST VACCINÉ?
1	15	1	Grosse Patate	chat	20	Boxer	70	noir	2012-03-24	2015-03-20	true
2	15	2	Jujube	chien	65	Labrador	52	blanc	2018-05-19	2017-10-23	true
3	15	3	petit Coeur	chien	21	chiwouawoua	30	petit	2016-03-21	2016-04-20	false

Pour la recherche, dans le champ de texte au-dessus de la table, on doit mettre une partie du nom de l'animal et appuyer sur le bouton "+". Ensuite, tous les tuples de les animaux qui ont cette partie de nom seront affichés.

Guide d'utilisations

1. Copier le fichier où il y a les schémas et coller le tout dans pgAdmin
2. Copier et coller dans pgAdmin les "data" afin d'insérer les tuples dans les tables
3. Exécuter le code SQL sur pgAdmin.
4. Donner les données pertinentes dans la fonction connectionConfig qui se trouve dans le fichier database.service.ts. Voici une exemple:

```
// TODO: A MODIFIER POUR VOTRE BD
public connectionConfig: pg.ConnectionConfig = {
```

```
user: "postgres",  
database: "clinique_db",  
password: "bib12345",  
port: 5432,  
host: "127.0.0.1",  
};
```

5. Maintenant avec le projet Angular, faites un “npm ci” dans le terminal dans le répertoire client et le répertoire serveur.
6. Ensuite, exécuter la commande “npm start” dans le dossier serveur et finalement dans le dossier client en roulant toujours le pgAdmin.
7. La configuration est terminée! L'application est supposée s'ouvrir adéquatement.

Conclusion

Ce fut une expérience enrichissante et amusante. Malgré, les nombreux échecs et tentatives lors du codage, le résultat était satisfaisant. Ce laboratoire est complet et nous a permis de mieux comprendre la matière. Il est suggéré de le refaire la session prochaine.